Language of Thought: The Connectionist Contribution

MURAT AYDEDE
The University of Chicago, Department of Philosophy, 1050 East 59th Street, Chicago, Illinois 60637, U.S.A. (email: maydede@midway.uchicago.edu)

Abstract. Fodor and Pylyshyn's critique of connectionism has posed a challenge to connectionists: Adequately explain such nomological regularities as systematicity and productivity without postulating a "language of thought" (LOT). Some connectionists like Smolensky took the challenge very seriously, and attempted to meet it by developing models that were supposed to be non-classical. At the core of these attempts lies the claim that connectionist models can provide a representational system with a combinatorial syntax and processes sensitive to syntactic structure. They are not implementation models because, it is claimed, the way they obtain syntax and structure sensitivity is not "concatenative," hence "radically different" from the way classicists handle them. In this paper, I offer an analysis of what it is to physically satisfy/realize a formal system. In this context, I examine the minimal truth-conditions of LOT Hypothesis. From my analysis it will follow that concatenative realization of formal systems is irrelevant to LOTH since the very notion of LOT is indifferent to such an implementation level issue as concatenation. I will conclude that to the extent to which they can explain the law-like cognitive regularities, a certain class of connectionist models proposed as radical alternatives to the classical LOT paradigm will in fact turn out to be LOT models, even though new and potentially very exciting ones.

Key words: Connectionism, Language of Thought, cognitive architecture, formal system, syntax, computation, implementation, concatenation, structure sensitivity, thinking, systematicity.

1. Introduction

Fodor and Pylyshyn's (F&P) (1988) critique of connectionism has posed a dilemma, and with it, a challenge to connectionists: adequately explain such cognitive regularities as systematicity and productivity without implementing a classical language of thought (LOT) architecture. Some connectionists took the challenge seriously and tried to meet the challenge by developing certain kinds of models that use distributed representations in certain new ways. These new ways are the basis of these connectionists' rejection of the first horn of the dilemma: namely, if connectionists, in their attempt to explain systematicity, postulate representations with syntactic (and semantic) structure and mechanisms that would process such representations in a way sensitive to their syntactic structure, then their models are fundamentally implementation models of LOT architecture. Therefore, F&P claimed, they have nothing new to offer since they fail to compete with classical models at the cognitive level.[1]

Minds and Machines **7:** 57–101, 1997.

The other, second, horn of the dilemma was that if connectionists don't postulate syntactically structured representations and structure sensitive processes, then they fail to adequately explain systematicity, hence their models are false as models of cognitive capacities that exhibit systematicity. Although many people have serious trouble with this horn of the dilemma too, I won't discuss it here. Instead I will restrict myself in what follows to the discussion of the problems raised by the rejection of the other horn.

The basic rational underlying the dilemma for F&P was that there are certain (law-like) regularities about (high-level human) cognition like systematicity that can be explained only by postulating a certain kind of cognitive architecture;[2] namely, one whose representations or data structures satisfy a certain description, call it **D**

According to F&P, the description **D** is any description according to which (cf. F&P, 1988: 12–13):

a. representations of a system have a combinatorial syntax and semantics such that structurally complex (molecular) representations are systematically built up out of structurally simple (atomic) constituents, and the semantic content of a molecular representation is a function of the semantic content of its atomic constituents together with its syntactic/formal structure, *and*
b. the operations on representations are (casually) sensitive to the syntactic/formal structure of representations defined by this combinatorial syntax.

F&P take (a) and (b) to be the *defining* characteristics of classicism or the LOT architecture (1988:13). This is why, I think, they seem to take the first horn of their dilemma as nonproblematic and don't discuss it at all in their article.

The situation is puzzling. For F&P, **D** is what makes a system a LOT system. If there are connectionist systems that genuinely satisfy **D**, as some connectionists claim, why do they not count as classical? What prevents them from becoming LOT systems? In other words, how is it possible for connectionists to reject the first horn of the dilemma? On the face of it, connectionists seem to be rejecting the definition given by F&P. If so, they must have a different conception of classicism, a different understanding of what the LOT Hypothesis (LOTH) involves.

Connectionists claim that it is not the mere satisfaction of **D** that is essential for LOT but rather it is how you satisfy **D** that determines whether a system is classical or not. Classicism, on this view, is essentially committed to *cancatenative* or *explicit* tokening of syntactic constituents of structured representations postulated in **D**-a, since, they claim, this is essentially how classicism envisages to obtain, as postulated in **D**-b, actual causal sensitivity in the processes run over syntactically structured representations.[3] Van Gelder, who has been the most outspoken defender of this view, puts the point thus:

> Classical theorists have a deep theoretical commitment to the idea that mental representations themselves are strictly concatenative while in Connectionist research an increasing tendency can be discerned to reject [explicit, concatenative] syntactic structure in the representations themselves in favor of an [implicit, non-concatenative] compositionality... [I]t can be seen how Connectionists

can use compositional representations, while at the very same time correctly claim to reject the traditional language-of-Thought hypothesis... [T]he most pertinent and informative contrast between the Classical approach and Connnetionism is. . . between two very different ways of *implementing* compositional structure. (1990: 365, my emphasis)

Van Gelder, along with many others, claims that simply postulating a representational system that satisfies **D** is not enough for the system to be a LOT system: **D** as such is not in the monopoly of classicists. Rather, it is how you implement **D** that matters. A LOT system is *essentially* one that *implements* **D** explicitly. Since some of the models developed by connectionists as a direct response to F&P's challenge propose to satisfy **D** only implicitly or nonconcatenatively,[4] they are not LOT models; hence the connectionist rejection of the first horn of F&P's dilemma.

On the face of it, this is puzzling too. For why should LOT be essentially tied to what seems to be such an implementation level issue as explicit realization of syntactic structure? The defenders of LOTH have always been very consistent and clear about their hypothesis being pitched at the cognitive level. But what is more puzzling is that Fodor and McLaughlin (F&M) seem to agree with connecionists that a LOT model is one that essentially satisfies **D** explicitly. In their (1990) article, they criticized Smolensky's Tensor Product Representations Model that Smolensky offered as a counterexample to the first horn of F&P's dilema. F&M make the following claim:

We... stipulate that for a pair of expression types El, E2, the first is a *Classical* constituent of the second only if the first is tokened whenever the second is tokened. (1990: 186, emphasis in the original)

Here they plainly require explicit realization of syntactic structure for it to belong to a classical representational system. It is not clear, however, to what extent they want to press on this requirement. In their criticism of Smolensky's way of incorporating syntactic complexity in the tensor product representations, they seem ambivalent: on the one hand, they seem to be willing to grant that tensor product representations do have constituent structure "in an extended sense" (1990: 200), while, on the other hand, they accuse Smolensky of confusing a representation's *having* constituent structure with its *representing* one. They claim that since the tensor product representations don't literally contain their constituents, they can at most represent syntactic structure but not have one. I will come back to this charge below (§4).

F&M's real worry however, seems to be **D**-b. They claim that since the syntactic constituents of tensor product representations are not tokened when the complex representations are tokened, they can't be causally efficacious. In other words, since the constituents are not actually there in the representations, the causal processes can't be sensitive to their constituent structure. Hence, their real criticism seems to be that explicit tokening of syntactic constituents is necessary precisely because without it **D**-b can't be satisfied. In a nutshell, F&M's criticism must, it appears,

ultimately come down to the claim that the proposed connectionist models don't satisfy **D** (since they don't satisfy **D**-b). In other words, they don't want to give away the antecedent of the first horn of their dilemma to connectionists who claim to have refuted it.[5]

However, when F&M require that explicit realization be necessary for obtaining structure sensitive processing, the nature of their claim is not clear. The modal force of the claim seems to be not logical but a nomological/empirical one. They don't argue for their claim except rhetorically by appeal to some sort of a "how else" claim, which seems to show that they regard it as self-evident, obvious.

Like many others, I am not convinced. But instead of arguing for one way or other, which seems to be a more or less empirical issue anyway, I will try to make, in what follows, a more general and conceptual point about the dialectic of the current debate as I set it up here. Contrary to what seems to be a consensus among the debating parties, I will show that explicit or implicit realization of syntactic structure is irrelevant to a proper understanding of classicism or LOTH: concatenative realization is an implementation level issue and as such should not be conceptually tied to what LOT essentially is. In this, my aim is to spell out clearly the minimal truth-conditions of LOTH by clarifying how LOT ought to be conceived. LOTH has been proposed as an empirical claim. But surely, in order to determine whether LOTH is true or false, we have to know what exactly it says, which is not entirely an empirical issue.

Let me be more explicit. I want to argue that satisfying **D** for a representational system is sufficient for it to count as a LOT system no matter how it is satisfied, i.e. whether with explicit structure or with implicit one. But of course, the very success of my argumentative strategy crucially depends on the possibility of there being genuine implicit satisfaction of **D** in its entirety. If there is no such possibility, my attempt to show what I want to show will be at best futile and at worst incoherent, depending on how you cash out the nature of the modality in question. It will be incoherent if it is *logically* impossible to satisfy **D** except explicitly. It will be futile and not very interesting if it turns out to be *nomologically* impossible to satisfy **D** except explicitly.

I think that the logical reading of the modality can't be sustained. So I set it aside.[6] How about the nomological reading? Is it really self-evident and obvious as F&M seem to presuppose? I have two points to make.

First, since the classicist's claim is only nomological and not logical, there are, the classicist accepts, logically possible worlds in which **D** can be satisfied implicitly. Then my argument, if successful at the end, should be taken to apply only to those worlds, which is, in fact, sufficient ground for me to make my conceptual point: explicit realization of syntactic structure is irrelevant for the very notion of LOT. Put differently, my argument, if successful, would show that in those worlds the representational systems that satisfy **D** only implicitly do still count as LOT systems, which goes to show again that the proper understanding of LOT does not involve such low-level requirement as explicit instantiation of constituents.

But secondly and more importantly, I would like to settle for a Scotch verdict with the classicist with respect to the epistemic status of the nomological reading of the impossibility claim. I believe that the recent developments in connectionist modeling have shown at least, if nothing else, that we don't know whether it is impossible to satisfy **D** except explicitly; in particular, we don't know whether it is impossible to obtain genuinely structure sensitive processing (**D**-b) without explicitly tokening the syntactic constituent structure. This is different from saying that the modal claim of F&M is false. For all we know, indeed, it may turn out to be true.[7] My point against he classicist is that we don't know that, not yet anyway. And this epistemic claim is easier to get, since its negation is never argued for by classicists. If this point is granted, there is at least the epistemic possibility that structure sensitivity be obtained without explicit structure in the way envisioned by connectionists (we will see the flavor of their proposals below). My argument in this paper then is that if implicit structure sensitivity turns out to be real, this would in no way show that such systems would fall outside of the classical LOT paradigm. So, as far as my argument is directed against the classicist (more specifically, against the requirement of explicit syntactic structure for LOT), I don't officially want to commit myself to there *actually* being connectionist models that presently satisfy **D** satisfactorily in its entirety in an implicit way.

Connectionists (some of them, anyway), of course, are not so shy about the claims they make with respect to the models they have been developing with an eye to meet the F&P challenge to adequately explain systematicity. Chalmers (1990, 1991) is quite straightforward in his claim to have empirically refuted F&P&M's claim by claiming to have produced an actual connectionist model that satisfies **D**-b without explicit structure, which I will describe below. Smolensky (1990a, 1990b, 1995) and van Gelder (1990), among many others, seem to think that even though there may presently be no actual connectionist models whose structure sensitive processes over implicitly structured vectorial representations are adequate to explain (inferential) systematicity the advances in actual connectionist modeling show that there is in principle no reason to believe F&M's modal claim. In fact, they seem to think that connectionist research has already shown the empirical possibility of implicitly obtained structure sensitivity.

Again, as far as my argument is directed against such connectionists, I don't want to enter into a polemic as to whether their models have indeed shown the falsity of F&M's model claim. As far as my purposes in this paper are concerned, I am willing to grant what they claim about their models, since granting this won't affect the point I want to make against them: if their models genuinely satisfy **D** implicitly, they are still LOT models.

Here is the plan of what follows. After briefly presenting some preliminary clarification in the next section (§2), I will take up the discussion of **D**-a and present two connectionist proposals about how to handle it with only implicit structure (§3). On the basis of an extended analysis of what formal systems are and what it is for physical systems to satisfy them, I will argue (§4) that to the extent to which they

technically/empirically turn out to be adequate, complex representations belonging to such non-concatenative connectionist symbolic schemes do indeed have genuine syntactic structure, contrary to F&M's claim, but for all that they are still within classical LOT paradigm.

Then, I will take up what seems to be the crux of the whole debate: can structure sensitivity be obtained with only implicit structure? I will first present an experiment conducted by David Chalmers (§5) who claims to have experimentally demonstrated that implicit structure can support structure sensitive processes. Then I will argue that even though there are serious technical difficulties in the way of a positive answer to this question, we still don't quite know the answer. But more importantly, we don't need to know the answer to see that a positive answer wouldn't mean a new paradigm radically different from the LOT paradigm. My strategy in arguing for this conclusion will consist in making very clear what exactly structure sensitivity is supposed to be (§6.2). And this will require a discussion of the explanatory significance of structure sensitivity. For his purpose, I will briefly rehearse (§6.1&7) some of the historical arguments given for LOTH just to see what notion of LOT, and in particular what notion of structure sensitivity they can maximally justify: if concatenation plays no role in these arguments for LOTH, we have a perfectly good reason for why concatenation is not an essential part of the notion of LOT.

Although this paper appears to be another contribution to the polemic between some connectionists and classicsts, as I hope will be clear as we proceed, my primary aim is more fundamental. My aim is to contribute to our understanding of what exactly LOTH is, which means in our context what it is for a physical system to satisfy **D**. As **D** has two parts, my strategy will have two parts. In §4, I will develop an analysis of what it is to satisfy **D**-a. I will show what the general principles are, and argue that they do not distinguish between implicit and explicit satisfaction of **D**-a. With respect to **D**-b, my strategy will be exactly similar. When we see what principles are underlying causal structure sensitivity (§6.2), we will see that they don't make explicit satisfaction of **D**-b any more genuine than the implicit one. Both are on a par theoretically. Given that historical arguments for LOTH are indifferent to any particular satisfaction of **D**, but rather require only **D**, however satisfied, the very notion of LOT, as I will argue, cannot be tied to explicit satisfaction of **D**. It is a by-product of this fundamental conclusion that some conntecionist models (to the extent to which they turn out to be technically/empirically adequate) are still within the LOT paradigm.

2. Some Preliminaries: Implementation and Cognitive-Classical Architecture

In their article, F&P examine the notion of a cognitive architecture. They think that a proper understanding of this notion is crucial for their criticism of connectionism since it is leveled against connectionist models *at the cognitive level*. They claim

that they don't have any quarrel with connectionist models proposed as implementation models of (classical) cognitive architecture. They characterize the notion of cognitive architecture as follows:

> The architecture of the cognitive system consists of the set of basic operations, resources, functions, principles, etc. (generally the sorts of properties that would be described in a "user's manual" for that architecture if it were available on a computer) whose domain and range are the *representational states* of the organism. (F&P, 1988: 10)

Their emphasis here is on what makes an architecture a cognitive one. But let us first focus on what an architecture is.

As suggested by the parenthetical remark, what F&P seem to have in mind here is whatever notion of architecture is involved when we consider current high-level computer programming languages like BASIC, PASCAL, PROLOG, LISP, etc. These languages have different architectures in that their syntax and organization (e.g., some may require ample use of "GO TO" statements, whereas others not, thus forcing the programmer to write highly "structured" programs), primitive operations (e.g., the square root function might be primitive in one but not in others), use of computational resources (e.g., memory, processor time), and the like, are different. In this sense, the architecture of these universal languages is indeed what is being described in their "user's manual" (e.g., when you buy an over-the-counter compiler for one of these languages).[8]

So, if the notion of a (computational) architecture is to be understood in this way, what makes it cognitive? What makes it cognitive, according to F&P, is that the primitive operations, functions, etc., of the architecture so understood have, as their domain and range, *representational* states, i.e., data structures (symbols) that, at a minimum, represent states of affairs in the world. So, an architecture is *cognitive* if and only if what is being processed in this architecture has such representational content.

F&P want to say, then, of any such cognitive architecture that it is *classical* if and only if **D**-a is true of what is being thus processed (i.e., representations) *and* the architecture does actually exploit the (syntactic/formal) structural features of the representations in processing them (hence, **D**-b).

It should be emphasized that what F&P define in terms of **D** is what it is to be classical for a cognitive architecture. Put differently, what they define is 'classical-cognitive.' This is important to keep in mind. For in citing **D**, they are not concerned with defining the predicate 'x is classical' *tout court*. That this is so is apparent from the fact that we may have a computational architecture with universal computational power that is not classical-cognitive in the sense defined, but that, nevertheless, may be used to implement any classical-cognitive architecture. For instance, simple universal Turing machines or von Neumann machines are just like that. Their basic architecture in many cases cannot be *classical-cognitive*, but nevertheless they can be used to *implement* any computational processes defined over representations that satisfy **D**-a.[9] Similarly, F&P allow that there may be

connectionist architectures that are not classical-cognitive but may nevertheless be used to implement architectures that are classical-cognitive.

The notion of implementation here is a technical one that needs to be carefully distinguished from the ordinary pedestrian one involved when we talk about the "implementation" of **D**. (As the attentive reader might have noticed I have so far avoided the term in describing my own argumentative strategy, and instead talked about satisfaction, realization, instantiation of **D**.) This technical sense derives its use from computer science according to which a program written, say, in PASCAL, is implemented, for instance, in the assembly language, which in turn is implemented in the machine code of the particular physical computer that happens to run it. On this picture, the architecture provided by PASCAL defines a virtual machine. It is virtual precisely because the actual physical hardware running it has a different architecture in the sense defined above. Implementation requires that there be a precise and complete mapping between the elements of computational architectures at different levels. Programming language compilers are in fact nothing but programs that effect such machine/hardware specific mappings from one level to the one at the bottom. In implementational hierarchies, a primitive operation of a higher-level architecture is usually implemented by a host of different and more primitive operations of the lower-level implementing architecture. Also, it is often the case that as you go down through the implementing architectures you lose the representational character and the precise structural organization of data structures of higher-level architectures.

The reason I am belaboring this point is that when I claim that connectionist models that satisfy **D** implicitly are still LOT models, I do not mean to claim that they are implementations of LOT models in this technical sense. In order to prevent any misunderstanding and minimize confusion I will generally avoid using the term in what follows, and use instead terms similar to 'instantiation', 'realization', etc. My primary aim, as I said, is to define what it is for any model or system to belong to the classical LOT paradigm.

Perhaps a more transparent illustration of **D** as the defining feature of classical-cognitive architecture can be found in the notion of an interpreted formal system. The proof-theoretic notion of a formal system consists of, first, constructing a formal language by means of an alphabet and a finite set of formation rules, and, then, of adding to this language a deductive apparatus (a set of derivation or transformation rules) that would define the rules of transforming the well-formed formulas of this language. The paradigmatic examples can be found in different formulations of propositional and first-order predicate logic.[10] In fact, it is no accident that there are strong parallels between **D**-a and formation rules given for formal *languages* on the one hand, and between **D**-b and the derivation rules given for formal *systems* on the other. To say that **D** is true of mental representations is just to say that they constitute (or, are characterizable as) an (interpreted) *formal system* in the logicians' or mathematicians' sense of the phrase – except for *causal* sensitivity, see below.

So, to recap, **D** defines what makes a cognitive architecture classical only by putting constraints on the nature of what is being processed and on the character of the processing in that architecture. To say, then, that any cognitive architecture that satisfies **D** is classical is just to say that the architecture processes representations with combinatorial syntax and semantics (**D**-a), and that the architectural mechanisms are so designed that they process the representations by (causally) responding to their formal/syntactic features defined by this combinatorial syntax (**D**-b).

It is very important to note that **D**-a and **D**-b are abstract meta-architectural properties in that they are themselves conditions upon any proposed *specific* architecture's being classical. There are in fact indefinitely many possible classical architectures. To illustrate the point, consider, for instance, different formulations of sentential logic: in one, the only formally complex sentences may be negations and conditionals in which case the transformation rules that are appropriate for *these* would define the primitive processing operations; in others, all the five standard logical forms of sentences and different sets of primitive rules for transforming them might be given. But, **D** would come out to be true of any different formulation of sentential logic if considered as a representational system run in a computational architecture. Similarly, any architecture (analogous to LISP, PROLOG, etc.) that would process such representations in a structure sensitive way would count as a classical one. This is the sense in which **D**-a and **D**-b are abstract meta-architectural properties. They define classicism as a *genus*, but *not any particular way* of being classical. LOTH as such, then, is not committed to any *particular* architecture or to any *particular* **D**-like representational system in advance. It simply claims that whatever the *particular* cognitive architecture of the brain might turn out to be, **D** must be true of it.

That **D** is a meta-architectural constraint is in fact the primary reason why it would be inappropriate to claim that any specific architecture that (implicitly or explicitly) satisfies **D** is an *implementation* (in the technical sense) of LOT architecture.[11] For, strictly speaking, there is no such thing as *the* LOT architecture in the technical sense of providing/defining a virtual computer. **D** does not provide such a notion. **D** only specifies what it is about a *class* of architectures that makes them all belong to a paradigm which we may characterize as the classical or LOT paradigm. In other words, it specifies what it is about them that unites them all under the class of classical-cognitive architectures. To repeat: my primary aim is to spell out what exactly it is to satisfy **D** so that successful satisfaction of **D** is seen to be sufficient for any model to qualify as classical. As will become clear later, I believe that if connectionists really succeed in satisfying **D** in its entirety adequately, this would be quite a remarkable and exciting contribution to our understanding of LOT. So I don't want to downplay their potential importance by saying that they ultimately belong to the classical paradigm.

3. The Connectionist Proposals

Now it is time to see how connectionists propose to satisfy **D**-a implicitly, i.e., how they propose to have syntactically structured representations without the constituents being part of the representation. This is, at any rate, necessary in order to evaluate the claims made about **D**-b. As I said, the real motivation that underlies the classicist insistence on explicit structure concerns structure sensitivity, hence **D**-b. I will eventually come to that. For the moment, I want to look into two proposals about how to satisfy **D**-a implicitly. My aim is just to convey the flavor of the proposals.

3.1. POLLACK'S RECURSIVE AUTO-ASSOCIATIVE MEMORY (RAAM)

Pollack (1990) has developed a connectionist architecture for a class of networks that can recursively encode tree structures with a fixed valence to an almost arbitrary depth. Since tree structures can be used to describe syntactic or formal constituents of expressions, any complex representation whose constituent structure can be analyzed by tree structures can be encoded in such networks. Since my aim is just to convey the idea, let me describe how the RAAM works on an example.

Suppose we want to produce connectionist representations of conjunctions. We can do so by using a RAAM network. RAAM networks use distributed representations, so we will need a set of vectors representing the atomic sentences which, of course, will correspond to activation patterns of some set of connectionist units in the network. In conjunctions, it is natural (but not necessary) to use 3-valence tree structures. So the basic architecture of the RAAM network we need will look like this.

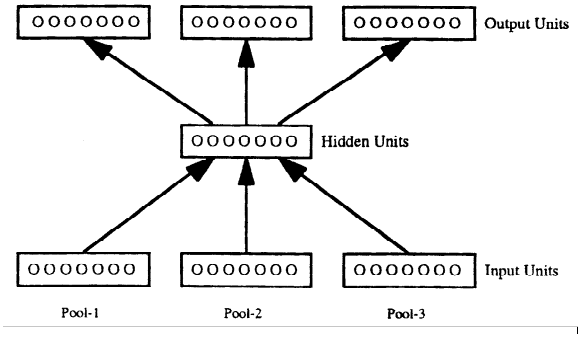

Figure 1. Pollack's Recursive Auto-Associative Memory (RAAM).

This is a feed-forward network: the activation spreads from the input units to the output units through the hidden units. Let us use 'A,''B,''C,' etc., to denote the vectors standing for atomic sentences. These will be fed to the units of Pool-1 and Pool-3, each vector to a single pool. These are the inputs to the network. The units of Pool-2, in each input cycle, will be fed by a constant vector, call

it &-vector. This vector is what makes the network encode conjunctions: it is a conjunction-marker. When the vectors for A/&/B are fed to the input units, the activation will spread to the hidden units creating a distinct activation pattern that can be treated again as a vector, which will in turn activate the output units. The aim is that the network should produce the same exact pattern in the output units as that of the input units. This can easily be done by using simple learning techniques. Since this is an auto-associator, unsupervised back-propagation learning method is natural here. When the network in this way learns to auto-associate the vectors A/&/B in the correct order in its output units given the same input, we will have a distinct activation pattern in the hidden units: the vector corresponding to this is the distributed connectionist representation of 'A&B' compressed to one third of the entire original input, and is non-concatentive, hence non-classical.

Notice two points. First, since this representation is produced by a process of auto-association, when it is supplied to the hidden units of the network, it will uniquely decompose to its original constituents in the output units. Second, the compressed representation can be resupplied as a conjunct to the input units of the network to produce yet another compressed, more complex, conjunctive representation. This is the recursive aspect of the RAAM architecture. When the network is suitably organized and large enough, the same network can produce, decompose, and store a very large number of conjunctions of almost arbitrary complexity. Furthermore, and this is the crucial point, the same network can be used to compose and decompose in the same manner any other complex representations whose "logical forms" are different. This can be done by replacing &-vector with other theoretically relevant vectors, for instance, with a suitably chosen v-vector, or even with a not-vector in which case one of the input pools will be supplied by a "nil-vector," and so on.

The RAAM architecture is still under development. Its prospects especially for natural language sentence parsing seem promising, hence it is natural to suppose that its impact on natural language processing as well as on any sort of analysis that requires variable binding in general will be significant.

3.2. Smolensky's Tensor Product Representations

In recent years, Smolensky[12] has developed a powerful connectionist technique for binding values to variables, all represented again by activity pattern vectors, hence using distributed representations. Although the technique, called 'tensor product variable binding,' is complicated, the idea behind it is simple. Let us use an example again.

Suppose we want to produce a tensor product representation corresponding to the sentence 'John loves Mary.' Since this sentence can be syntactically decomposed into its constituents, we can work on its syntactic structure:

$$\{(\text{John})_{\text{NP}}[(\text{loves})_{\text{VP}}(\text{Mary})_{\text{NP}}]_{\text{P}}\}_{\text{S}}$$

The sentence (S) is first decomposed to a noun phrase (NP) and a predicate (P), then the predicate is decomposed to a verb phrase (VP) and a noun phrase. These can be taken to be syntactic "roles" or "positions" (variables) that need to be filled by particular lexical items: in our case the "fillers" (values) are 'John,' 'loves' and 'Mary.'

Smolensky postulates a set of particular filler-vectors (these are the connectionist representations corresponding to lexical items), and a set of particular role-vectors for syntactic positions (for instance, an NP-vector, a P-vector, a VP-vector, and so on). If we want to bind a filler vector to a role-vector, say, the vector representing Mary to the NP-vector, we multiply the two vectors to get their tensor product, the result is the tensor product vector representation for 'Mary' in the NP-position We then perform the same operation for 'loves' in the VP-position. We then superimpose the resulting two vectors (i.e., add the two vectors by simple vector addition) to get a new filler vector to be bound to the P-vector. When we do this we have a tensor product vector for the predicate bound to particular values. After similarly binding (by tensor product operation) the vector representing John to the NP-vector, we can now get a single vector corresponding to the whole sentence by simply superimposing the two vectors, namely the NP-vector bound to "John" and the P-vector bound to, as it were, "loves Mary." This is the compressed distributed connectionist vector representing the state of affairs [John loves Mary].

Also, under certain conditions, there is, Smolensky claims, a connectionist network which will uniquely decompose it back to its constituents.[13] This connectionist representation does seem to have constituent structure. But, again, it is non-concatenative. Notice also the recursive aspect of this technique: the tensor product vectors (vectors standing for roles bound to fillers) can be re-used as fillers to be bound to further roles, as we have just done in binding the P-vector. By using the same technique, we can still bind the whole vector corresponding to the sentence in question, say, to a left-hand-conjunct-vector (a role vector) in order to get a new vector representing the state of affairs, say, [John loves Mary and Mike hates John] and so on.

There are other attempts to develop techniques for incorporating complex distributed representations into connectionist models.[14] All of them use similar procedures to capture syntactically structured representations in the form of compressed vectors, i.e. implicitly. And all of them are committed to distributed representations. In a sense, this is no surprise, since resources, especially the number of processing units, in connectionist networks are limited. For this reason, connectionists had to find out ways of using finite resources over and over again in a recursive fashion in order to handle, to a psychologically respectable degree, the problems posed by what *prima facia* seem to be recursive cognitive capacities.

4. Formal Systems and Their Instantiations

F&M's article was a response to Smolensky's Tensor Product System. As I have mentioned above, they accuse Smolensky of confusing two issues that need to be clearly distinguished. Namely, Smolensky, they claim, confuses the issue of a representation's actually *having* syntactic structure with the issue of a representation's *representing* syntactic structure.[15] F&M claim that Smolensky's tensor product representations do only the latter; such representations do not themselves *have* any actual syntactic structure. This issue relates to whether there can be explicit satisfaction of **D**-a. Even with respect to **D**-a, then, F&M seem to think that for genuine syntactic structure explicit instantiation of constituents is necessary. I believe that this claim (with respect to **D**-a) is false.

I will argue against it by considering the minimal conditions that need to hold in order for a formal system to have a notation.[16] Since there is a clear parallel between providing a notation and a physical instantiation in a machine (or, organism), my discussion will equally apply to effecting a physical instantiation mapping from an abstractly characterized formal system onto the states of a physically realized (computational) machine. In what follows, I will leave the issue of semantic interpretation aside, and assume that the formulas in question are to be semantically interpreted. As far as we keep in mind that semantic interpretation is not an issue that divides classicists and connectionists (at least the ones under consideration here – both camps are intentional realists that accept the reality of representational states *qua* representational), there is no harm in focusing only on formal issues. This will also make the exposition more easy and tractable.

We need to remind ourselves that formal systems are abstract entities. By this, I simply mean that for their existence no *particular* notation is necessary. There is something about formal systems, in other words, that in some interesting sense transcends their notational realizations. There are many quite different kinds of formal systems. But the ones we are interested in are the ones whose structure conforms to **D**, i.e. ones with combinatorial/recursive rules. Sentential Logic (SL) is a prime example of such a formal system. It will be easier to make my point on a concrete example. Let us then work on the example SL provides. Here is an abstract characterization of SL with only three logical forms (conjunction, disjunction and negation).

ABSTRACT CHARACTERIZATION OF SL

I. There is a set of distinct/disjoint *atomic* sentences in the language of SL.

II. **Formation Rules for sentences of SL:**

(1) Each atomic sentence is a *sentence*;

(2) For any x and y, if x and y are sentences, then there are three (formative) operations N, C, D, such that N(x), C(x,y), and D(x,y) are (non-atomic) sentences;

(3) Nothing else is a sentence in SL.

- **REMARK 1**: (Terminology) N(x) is the *negation* of x. C(x,y) is the *conjunction* of x and y. D(x,y) is the *disjunction* of x and y. x and y are called *conjuncts* in C(x,y) and *disjuncts* in D(x,y). Any output of any operation is a *complex* sentence. Any sentence that is an argument to any operation is a (syntactic) *constituent* of the output complex sentence. (The sentences mentioned in I and II are, of course, sentence *types*.)
- **REMARK 2**: (Conditions on Formation Rules) The formative operations in (II.2) are such that: for any x and y, if x and y are sentences or 2-tuples of sentences, [17] then for any operation Ω and Ψ,
  (4) $x \neq \Omega(x)$;
  (5) $x = y$ if and only if $\Omega(x) = \Omega(y)$;
  (6) $\Omega = \Psi$ if and only if $\Omega(x) = \Psi(x)$;
  (7) if $\Omega \neq \Psi$, then $\Omega(x) \neq \Psi(y)$;
  (8) Ω is an effectively computable function such that there is an "inverse" operation Θ such that Θ effectively computes $< x, \Omega >$ given $\Omega(x)$.

III. **Transformation Rules:**

[ADJ] Given any two sentences, derive their conjunction.
[CON] Given any conjunction, derive any one of its conjuncts.
[ADD] Given any one sentence, derive any disjunction one of whose disjuncts is the given sentence.
[DS] Given a disjuction and the negation of one of its disjuncts, derive the disjunct.
Etc.

This characterization of SL is abstract in the sense that it is notation-free. It puts constraints on any notation that would aspire to be a notation *of* SL. An indefinite number of notational schemes can satisfy this abstract characterization. Put differently, what makes indefinitely many notations equivalent (hence, notations *of* SL) is the existence of systematic ways of satisfying the above abstract characterization. So, let us start by specifying what it takes to provide a notation for SL.

To begin with, notice that the abstract characterization of SL is, intuitively, a characterization of a "digital" system: the atomic sentence *types* (what Goodman calls 'characters') of any proposed specific notation are stipulated to be distinct from each other, i.e. they must be syntactically disjoint, to use Goodman's expression. (The conditions in REMARK 2 are meant to transfer this disjointness to complex sentence types; they are meant to guarantee the uniqueness of complex types – see below.)

Providing a specific notation for formal systems generally proceeds through two major phases. The initial phase is to concretely specify the atomic symbols, in the case of SL, the atomic sentences. How this is done?

Preserving syntactical disjointness requires:

- providing an identity criterion for each type the satisfaction of which by the tokens (inscriptions, marks) is both necessary and sufficient for the tokens to be *of* (or, belong to) a certain type such that

- no token can be *of* (belong to) more than one type.

In other words, the identity criteria express the "essences" of certain abstract types (kinds): they are what define being of a certain type. And since the types are stipulated to be distinct, the criteria for types must be such that anything that satisfies any criterion must ipso facto fail to satisfy other criteria. Types are, therefore, as Goodman would put it (1976:132–3), abstraction classes of type-indifference among tokens.

An identity criterion in itself may consist of a disjunctive set of quite heterogeneous (and, even arbitrary) conditions or elements. What is important is that it should succeed in defining an equivalence-set for tokens such that no token satisfying the criterion would satisfy any other (i.e. would belong to any other equivalence-class, therefore to any other type).[19]

Tokens are physical spatio-temporal particulars. As such, it is not quite a straightforward task to provide a specific notation that would meet all the above conditions. The identity criteria for types must be so chosen that the determination (recognition/reading) and production (copying/writing) of tokens must be, in Haugeland's terms (1982: 214), *positive* and *reliable*. A positive determination/production procedure is "one which can succeed absolutely and without qualification" (214), and a reliable procedure is one "which, under suitable conditions, can be counted on to succeed virtually every time" (215). Although it may be difficult to come up with such procedures, providing identity criteria for guaranteeing such procedures is a matter of the imagination of the formalist, and for that matter, of the computer engineer since the physically built computers have states that are, under suitable interpretations, revealed to be symbolic states. Preserving the type-identity of physical states under an interpretation mapping is essential to their working, and indeed to their being computers (see below).[20] Specifying identity criteria for atomic types that would satisfy the aforementioned constraints is, then, what it means to specify *concretely* the atomic symbol types.

In logic textbooks and courses, specification of a formal system is not usually distinguished from providing a notation for it; thus, what is essential (the abstract structure) and what is accidental (the particular notation provided) are not usually distinguished. These two issues are run together. Specifying the atomic types concretely is standardly done by supplying a token for each type with the hope that the tokens will give enough idea of what the types are. As I said, tokens are physical entities, and as such they have certain physical properties. By providing token for each type, we in fact try to indicate that certain physical features of the tokens are what makes the tokens tokens of a certain type. Thus we use tokens as identifying examples of their types. In other words, we identify the primitive types by ostension. Here is one way it would go for atomic sentence types of SL:

Atomic sentences of SL : 'A,' 'B,' 'C,' 'D,'...

The point to emphasize here is that in providing a notation the atomic symbol types are individuated by certain sets of (quasi-)physical properties of their tokens. Any

token produced to satisfy a certain set of physical properties, say, a certain shape, is a token of a particular atomic symbol type.[21] In this kind of implicit procedure, this is what it is to provide identity criteria for atomic symbol types.

The second phase in providing a notation is to specify the formative operations concretely. Since the formative operations are what define the syntactic constituency relations among symbols, what needs to be specified concretely is "a mode of combination" for symbols.[22] This mode of combination must not only satisfy the conditions in REMARK 2 but also reflect their recursive character. Here is a standard example that does both in the case of SL:

$$\begin{aligned}
&\text{Operation N} : \mathrm{N}(x) \quad = \text{`}\sim\text{'}\hat{}\text{`}x\text{'} \\
&\text{Operation C} : \mathrm{C}(x,y) = \text{`(`}\hat{}\text{`}x\text{'}\hat{}\text{`\&'}\hat{}\text{`}y\text{'}\hat{}\text{`)'} \\
&\text{Operation D} : \mathrm{D}(x,y) = \text{`(`}\hat{}\text{`}x\text{'}\hat{}\text{`}\vee\text{'}\hat{}\text{`}y\text{'}\hat{}\text{`)'}
\end{aligned}$$

where *x* and *y* are any (atomic or molecular) sentence and ' ^ ' is meant to be the concatenation symbol. Now that the atomic symbols are concretely specified in the way indicated above, any substitution instance of the formation operations so specified will give us a (syntactically) complex sentence. Also, notice that since we now have the concretely specified modes of combination, we have two kinds of individuation criteria: one for the particular sentence types with which we can distinguish, for instance, between '(A&B)' and '(C&D),' and one for the logical type (form) of sentences with which we can distinguish between negations, conjunctions, and disjunctions, e.g. between '(A&B)' and '(A∨B).' (This distinction between logical forms, like disjunctions, conjunctions, etc., and logically identical types with different constituents will be important when we come to discuss structure sensitivity below.) Clearly, this standard scheme just indicated does satisfy the conditions specified in REMARK 2.

The significance of these conditions, *inter alia*, is that they ensure the *uniqueness* of the output of operations given distinct input and the *constancy* (or the sameness) of the output given the same input. Compliance with condition (5) guarantees two things: the procedures for forming a complex sentence and then decomposing it back to its constituents (thereby making its logical form explicit) are mechanically realizable. In short, what these conditions together guarantee is the syntactic disjointness of *complex* sentence types together with positive, reliable and effective procedures for producing and identifying them. Put differently, when the modes of combination satisfy the conditions in REMARK 2, they will provide identity criteria for complex sentence types, and these criteria will be such that they will not only secure the syntactic disjointness of complex types but also will guarantee the recoverability of syntactic constituents down to the atomic types by recursive and effective procedures. Specifying procedures for combining (already concretely specified) atomic types that would satisfy the conditions in REMARK 2, then, is what it means to specify *concretely* the mode of combination.

The notational scheme I have just provided is more or less the standard concatenative one. But in fact there are indefinitely many others. Almost all the familiar

notational schemes use what is called a concatenative mode of combination in their concrete specification of the formative operations. Let me be more precise:[23]

A mode of combination is *concatenative* if and only if when a syntactically complex symbol is tokened, some quasi-physical aspects or features of the token satisfy the individuation criteria for typing all the token syntactic constituents of it.

Intuitively, in concatenative schemes, any token of any complex symbol type contains, literally and explicitly, the tokens of its proper constituents, such that when a token of the complex symbol is produced, the tokens of its constituents are produced too. As we have seen, defined this way, concatenation is what F&M mean by "classical constituent" (see the quotation above). For instance, certain (spatial) parts of the token '(A&B)' satisfy the individuation criteria for its constituents, namely 'A' and 'B,' which are given in the first phase while concretely specifying the atomic sentences of SL.[24]

Providing a concatenative notation of SL, is only one way and not necessarily the only way of satisfying conditions in REMARK 2. True enough, it is the most practical one. But, in principle, there is no theoretical difference between concatenative and a non-concatenative notational schemes, *in so far as the scheme satisfies the conditions in REMARK 2*. These conditions put no theoretical requirements on whether the instantiation of abstractly characterized SL be a concatenative, or for that matter, non-concatenative one. They don't differentiate between such different instantiations. We may in fact think of the formative operations as simple input/output devices or little black boxes, so that when you supply the inputs they output further complex sentences. The only constraints on these devices is that they should comply with the conditions of REMARK 2.

As noted by many people, one good example of a non-concatenative instantiation scheme is the Gödel numbering procedure used in encoding the expressions of a formal language. This procedure uses a quite effective method to assign to each well-formed expression of a formal language a unique natural number. And it does this in a recursive manner. First, a distinct natural number is assigned to each of the primitive expressions of the language. Then the formative operations are specified by a distinct set of prime numbers. When the numbers standing for expressions are supplied, the operations produce (by using certain simple mathematical operations on both the supplied numbers and the prime numbers characteristic of each formative operation) a unique natural number standing for a complex expression whose constituents are the initially supplied number-coded expressions. (We may think of the "boxes" or I/O devices, which concretely specify the formative operations of SL, as embodying the necessary operations over numerals.) Using terms like 'expressions encoded in numbers' might make the Gödel numbering scheme appear as somehow parasitic upon concatenative schemes. But this is not necessary. You can think of the symbols of the language as consisting solely of numerals, and the operations of the formal language as operations over these numerals. What is truly remarkable about "Gödelese" is that thanks to the theorem of prime decomposition there is an effective decomposition procedure by means of which we can uniquely

recover the constituents of any given complex Gödelese expression and also identify its logical form. Gödelese is not a concatenative scheme: complex Gödelese expressions, when tokened, do not literally contain the tokens of their constituents, except by accident.

Other examples of non-concatenative instantiation schemes seem to be provided by the kind of connectionist representational schemes we have seen above. Let us take up Pollack's RAAM, and think of the concrete specification of formative operations C, D, and N as the specification of little boxes or I/O devices. When we specify concretely the mode of combination, we in fact provide a specification of the internal workings of these boxes. In this vein, we may think of the RAAM network as the concrete embodiment of these devices. You supply connectionist distributed representations as input, the device outputs a complex representation. For instance, think of Pollack's RAAM architechure as a concrete specification of Operation C when we supply &-vector to Pool-2, or Operation D when we supply v-vector, or Operation N when we supply $\sim$ -vector (with the nil-vector). Pollack, it appears, does exactly what is needed to be done in satisfying the abstract characterization of SL. He first concretely specifies the atomic sentences by providing individuation criteria for them in just the required sense. He indicates what count as the atomic sentences. They are concretely specified vectors whose disjointness is non-problematically obtained. The second phase is completed by concretely specifying the mode of combination recursively defined over these. And this is the RAAM network itself, or its complete mathematical description thereof in terms of vector algebra.

But do the suggested modes of combination really satisfy the conditions in REMARK 2? Or can they, within a similar connectionist framework?

A correct answer to this question requires a thorough examination of the technical characteristics/capacities/limitations of such networks which I cannot take up here. Supposing that syntactic disjointness for atomic vectorial symbols can be ensured, the proposed ways of dealing with combination procedures must secure syntactic disjointness for arbitrarily long and many complex symbol types in such a way that recovery of constituent structure should be positive and reliable. It is not at all obvious that these kinds of models can do that. I will, however, simply ignore this point for the moment and assume a positive answer to the question here, which seems to be rather an empirical issue. For I am trying to make a conceptual point: supposing that these models can technically face up to the job successfully, what follows? Notice that it is not this kind of worry that F&M have in mind when they object to Smolensky's tensor product representations: at least for polemical purposes, they seem to be assuming that syntactic disjointness can be reliably and effectively secured by Smolensky's techniques. I assume the same here.

Now that we have an analysis of a formal system and what it is for notations (physical systems) to be instantiations of a given (abstractly characterized) formal system, what can we say about F&M's accusation? Recall that they claim that only symbols that belong to concatenative notational schemes can genuinely have

syntactic constituent structure, the rest are schemes whose symbols can at best represent syntactic structure but not have one. So they seem to believe that non-concatenative notational schemes are somehow not genuine schemes. But what might be the basis for this belief? Given that the conditions I specified for providing a notation do not differentiate between concatenative and non-concatenative notations, I can see no reason other than a question-begging one: simply stipulate that only concatenative schemes do have genuine syntax, claim that the rest are bogus, and criticize connectionists accordingly.

Let me briefly recapitulate. Formal systems are abstract systems. There are certain conditions that need to be met by any concrete instantiation (notation or physical realization) of a formal system. I spelled out what those conditions are by using the example of SL. These conditions do not differentiate between concatenative and non-concatenative instantiations of formal systems. Therefore, if one scheme is a genuine instantiation of a formal system, so is the other. *A fortiori*, if a complex representation belonging to one scheme does genuinely have a syntactic structure and constituents – as opposed to representing the structure – so does the one belonging to the other scheme. Hence, as far as **D**-a is concerned, F&M cannot have any good argument for pressing that concatenation is necessary for genuine concrete instantiations of formal systems. Of course, as I said, their real reason for pressing for explicit structure is structure sensitive processing. They think that structure sensitive processing requires concatenation, to the discussion of which I now return.

5. Connectionists on Structure Sensitivity

Can non-concatenatively structured connectionist representations engage in structure sensitive processes? In other words, can connectionist models genuinely satisfy **D**-b?

The general consensus seems to be that if connectionist models using non-concatenative compositionality have to first decompose the compressed complex representations back to their constituents, thereby making their logical form available, in order for the structure-sensitive processes to operate on them, then the models are rightly to be called LOT models.

Many conectionists,[25] however, have proposed that connectionist models using some non-concatenative composition technique can directly process structurally complex representations in a structure-sensitive way without first decomposing them into their constituents, i.e., they can operate on non-concatenatively compositional representations *holistically*, as it is sometimes called. And, it is claimed, it is *this* feature of connectionist models that makes them at bottom truly and radically non-classical.

As we may remember, in their reply to Smolensky, F&M seem pretty confident that structure sensitive processing, hence inferential systematicity, can only be

guaranteed in a concatenatively realized scheme. Here is the only "argument" for this claim I was able to find:

> The relevant question is... whether [tensor product representations] have the kind of constituent structure to which an explanation of [inferential] sytematicity might appeal. But we have already seen the answer to *this* question: the constituents of complex activity vectors aren't "there," so if the causal consequences of tokening a complex vector are sensitive to its constituent structure, that's a miracle. (1990: 200)

As I said, it is not clear what the nature of the claim is. It seems that F&M take it to be a self-evident empirical truth. Here is how it *could* be false.[26]

Chalmers' Experiment. Chalmers (1990) has conducted a toy experiment which shows in a compact way how connectionist models might be able to handle syntactic transformations by operating holistically on connectionist complex representations. First, by using a RAAM architecture exactly similar to the one I described above, he encoded 125 active English sentences that are permutations on 5 proper names and 5 transitive verbs, and their passive forms, totaling 250 sentences. Chalmers then trained a simple three-layered feed-forward network to associate 70 active sentences with their passive forms (see Figure 2). Then, when he supplied the remaining 55 active sentences to the network one by one, the network produced their proper passive forms. Since these were in compressed form like the active ones used as inputs, he then supplied the outputs to the decomposing network. All of them correctly decomposed to their constituents in the right order. The success of the generalization of the network was 100%. He experimented also with first supplying the compressed passive sentences into the transformation network to get their active form. The results were equally successful.

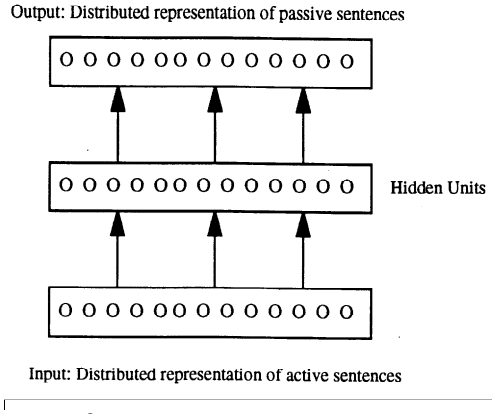

Figure 2. Transformation Network.

There are various important points about holistic connectionist processing that come out nicely in Chalmers' experiment. The most important one is the astonishing success rate of generalization of the transforming network. Let me emphasize

what we have here. We have a bunch of compressed vectors that are the connectionist representations. Furthermore, these representations are non-concatenatively complex (§3.1). They have syntactic structure, as we have seen, in one well-defined sense (§4). The transforming network is trained to process these in a certain way that is determined by, say, an abstract (interpreted) formal/syntactic system.[27] When the training is complete, the network acquires a general capacity to transform similarly structured representations in the appropriate way. The success of the network's generalization over vectors for which the network is not trained is clearly not accidental. This quite robust generalization rate of the network seems to make a strong case for the claim that structurally similar connectionist representations are processed in similar ways, i.e., as their logical form requires.[28] The generalization success of the network makes it quite clear that structure sensitive processing is *non-accidentally* obtained for all the structurally similar complex representations: i.e., nothing similar to look-up tables or brute force storing exists. Clearly the network somehow learns to detect the form of the complex representations supplied and process them as their form requires. What does this mean?

Well, Chalmers claims that it just means that non-concatenatively complex representations *can* be processed in a structure sensitive way just as **D**-b requires, and therefore, he hastens to add, these results experimentally refute the modal claim made by F&M. In other words, he takes himself to provide an empirical refutation of the claim that structure sensitivity can only be obtained with concatenation. This is the basis of his claim (along with many others') that such connectionist models provide a radically different alternative to LOT models.

Is Chalmers right? Can he be right? Hard to tell. Part of the reason why we do not know how to settle the epistemic issue of whether the modal claim is true is that we do not know what exactly the truth-conditions of the claim are in the first place. Let us then get clearer about what structure sensitivity really comes to so that we can evaluate the important claims made on behalf of connectionist models that are supposed to provide a radical alternative paradigm to that of LOT.

6. Structure Sensitivity and LOT

In this section, my aim is to spell out very clearly and explicitly what structure sensitivity is, i.e., what it means for a process defined over representations to be sensitive to the syntactic structure of those representations. I also want to say exactly what the explanatory significance of postulating structure sensitive processes is. In other words, I want to clarify what is theoretically so exciting about structure sensitivity: what is the problem or task it is supposed to solve or accomplish? Answering this question, especially in its historical context, is crucial to evaluate connectionists' claims.

The intuition I want to exploit and ultimately show to be correct is that there are general principles that are in play whenever the representational processes of a system (device, organism) are structure sensitive. I want to unearth those principles

and show how they are supposed to constitute an exciting solution to a set of problems, i.e. how they are supposed to explain certain set of phenomena. Once we see what those principles are and how they figure in the solution of the problem, we will be in a position to see that connectionist holistic processing conforms to those principles (to the extent to which such holistic processing can technically face up to the demands of the job – see below). This is exactly what needs to be done if we are to evaluate whether non-concatenatively obtained structure sensitivity provides an alternative to LOT framework.

6.1. The Traditional Arguments for LOTH

I will start with some general remarks about the arguments or reasons standardly given for LOTH. As I said, my aim is to see what these arguments at most require regarding the very notion of LOT *vis-à-vis* its explananda. They have the following general form. They all point to certain cognitive phenomena. Certain features or regularities in these phenomena are carefully noted and detailed. And finally, it is argued that a representational systems that satisfies **D** is the best explanation of these phenomena, and inferred that LOTH must be true. These arguments are therefore not meant to be apodeictic or demonstrative. But rather, postulation of LOT is justified as an inference to the best available explanation of the cited cognitive phenomena.

The set of cognitive phenomena in question can be grouped into two: on the one hand, there is the set of what I will call *formative* regularities, and on the other hand, *transformational* or *inferential* regularities. Intuitively, formative cognitive regularities are all those whose explanation essentially draws on exploiting only **D**-a in the postulation of LOT. For instance, productivity of thought, namely, the alleged empirical fact that a normal adult person can in principle entertain an infinite (or at least indefinite) amount of thoughts with arbitrary complexity, is a formative regularity in this respect: its classical explanation requires postulating a set of lexical/atomic representations with a combinatorial/recursive process of combining them (mental grammar) to obtain new representations, hence a LOT with at least **D**-a.

Fodor's discussion of systematicity of thought can again be split up into two: *formative* and *inferential* systematicity. Formative systematicity is the alleged empirical fact that the capacity to entertain a thought, at least for normal adult people, is intrinsically connected to the capacity to entertain certain other thoughts whose descriptions can be obtained by permutations over syntactically salient parts of the description of the original thought.[29] For example, the capacity to think that John loves Mary – is said to be intrinsically connected to the capacity to think that Mary loves John. Any organism who is genuinely capable of entertaining the one will be capable of entertaining the other. The classic explanation of this cognitive phenomanon draws again on postulating a representational system of which at least **D**-a is true.[30]

In fact, it seems that an even weaker hypothesis, weaker than postulating a language with recursive syntactic (and semantic) rules, can explain formative cognitive regularities. The British empiricists, for instance, thought that thoughts consist of entertaining ideas, which they modeled after pictures (that was how they attempted to account for the intentionality of ideas). Ideas are, of course, mental representations. If you can have a way of combining simple ideas into complex ones in some principled fashion, you may be in a position to explain formative regularities without postulating a *syntax* defined over linguistic/lexical items. Putting, for instance, simple pictorial atoms together, so to speak, to obtain complex pictures (as in the case of obtaining the idea of a unicorn, for instance) does not seem to require a full-blown syntactic apparatus, not at least in principle. But for that very reason such proposals may not be explanatorily adequate: the difficulties of the empiricist theories of concept and thought/thinking are notorious. It is, as F&P (1988: 49–50) note, precisely because many British empiricists, for instance, didn't have a syntactic story to tell in their account of productivity that they had no plausible story to tell about inferential thought processes. As far as thought processes were concerned they were all associationist: they had to appeal to statistical rather that structural properties of ideas.

The reason I brought this issue up is not because I believe that these empiricist stories are adequate in practice, but because I want to show clearly *what parts of LOTH* are meant to explain *which kinds of phenomena* and what is essential in this explanation. What seems to be essential in the explanation of formative regularities is the postulation of atomic representations and *some kind of* recursive apparatus to combine them to obtain more representations. In principle, there doesn't seem to be a necessity that this recursive apparatus be syntactic.[31] That this apparatus had better be syntactic is a demand that forces itself when one wants to give an adequate account of inferential/transformational regularities of cognition. Syntax seems to be a *processing* demand.

Notice that if connectionists' attempts to provide implicitly structured representations technically succeed, i.e., if they genuinely satisfy **D**-a with their non-concatenative syntactic constituents, they are in a position to genuinely explain the formative cognitive regularities like productivity and formative systematicity. I have argued above that if their models turn out to be technically adequate, such connectionist representations do genuinely have syntactic structure. But this genuine success is only partial. They must do equally well for the explanation of *inferential* cognitive regularities.

The cognitive phenomena that I call *inferential* (or, transformational) concern representational *processes*. For instance, cognition is systematic not only with respect to capacities to entertain thoughts but also with respect to inferential capacities, i.e. capacities to *process* thoughts. What I call *inferential* systematicity, in this respect, is the alleged empirical fact/regularity that the capacity to make a certain inference is intrinsically connected to the capacity to make certain others. Again, to use an example, the capacity to infer A from A&B is said to be intrinsically con-

nected to the capacity to infer A&B from A&B&C. If you are capable of engaging in one, you are *ipso facto* capable of engaging in the other, and in fact, many others involving conjunctions. The classical explanation of this phenomenon draws on **D**-b, i.e. the ability to process representations according to their syntactic structure. If representations can have common syntactic structure like being a conjunction, then operations can apply to them in virtue of their being conjunctions.

Although recent discussions of LOTH tend to focus on the need for a proper explanation of productivity and especially formative and inferential systematicities of cognition, it is absolutely essential to realize that these phenomena are not the only, or even the most important, grounds that can be used as arguments for LOTH. In fact, the single most important argument for LOTH, both historically and theoretically, has been, and still is, its prospects of offering a solution to what we might call "the problem of thinking."[32] Fodor has been insisting on this for years at least since the publication of his influential book *The Language of Thought*.[33] In fact, LOTH (spelled out simply as the existential hypothesis that *there is* a system of representations realized in the brain of sufficiently sophisticated cognitive organisms and that this system satisfies **D**) is regarded by Fodor as the only plausible story to tell about how to solve one of the three greatest mysteries of human mind:

> How could anything material have conscious states? How could anything material have semantical properties? How could anything material be rational? (where this means something like: how could the state transitions of a physical system preserve semantical properties?). (1991: 285, Reply to Devitt)

He took the LOT story to solve the third one, what I call the problem of thinking.[34] The computational picture of the mind/brain that is involved in LOTH (**D**) offers a naturalist solution to the problem of explaining how thinking, understood dynamically as a thought process, is possible assuming that thoughts are already understood as intentional, i.e., representational, states of the brain.[35]

Some[36] have even gone so far as to suggest that a proper understanding of what thinking is *logically* entails LOT! In other words, empirically inferring to LOT, as Fodor does, as the best available explanation of how thinking is possible is just too weak for these authors. The mode of inference must in fact be stronger. LOTH is not a "merely empirical" hypothesis (Rey), and can be established on "*a priori* grounds" (Davies). To justify such a strong claim, they maintain, all we need to do is to heed what thinking really is. I don't think that this stronger claim is right, but it highlights very nicely, for our present purposes, the strength and significance of the phenomenon of thinking in arguing for LOTH.

The purpose for belaboring this point is that the classical explanation of thinking as a specific sort of thought process essentially draws on postulating processes that are causally sensitive to the syntactic structure of representations, i.e. on appealing to **D**-b. Given that **D**-b requires **D**-a and that offering a solution to the problem of thinking is the single most important argument for LOTH, if we can see what

general principles are involved in the solution offered (i.e., how exactly structure sensitivity is supposed to explain the phenomenon of thinking), perhaps we can see that the very same principles are involved in the explanation that connectionists offer via an appeal to non-concatenatively obtained structure sensitivity. If we can do that, however new and otherwise exciting may be to obtain structure sensitivity in this sort of way, i.e. non-concatenatively, we would see that such connectionist models still fall within the classical LOT paradigm. I will of course argue for exactly this conclusion. So let us see, first, what it is about thinking that makes it hard to explain within a naturalistic ("mechanistic") framework. I will be brief since it is already discussed and described by others in better and more eloquent ways.[37]

Thinking is *at least* the tokenings of states that are (a) intentional (i.e. have representational content) and (b) causally connected. But it is more. There can be a causally connected series of intentional states that makes no sense at all. Thinking therefore is causally proceeding from states to states that would make a semantic sense: the transitions among states must preserve some of their intentional properties. In the ideal case, this property would be the truth value of the states. But in most cases, any interesting intentional property like warrantedness, degree of confirmation, semantic coherence given a certain practical context like satisfaction of goals in a specific context, etc. would do. In general, it is hard to spell out what this requirement of "making sense" is. The intuitive idea, however, must be clear. Call this general phenomenon the "semantic coherence" of causally connected thought processes. But thinking is still more. For you can have causally connected state transitions that would make semantic sense, but nevertheless wouldn't count as thinking. For example, any scenario under which a series of semantically coherent state transitions would be causally connected to each other but "in the wrong sort of way" would illustrate the point. There are some very nice illustrations of this kind of scenario in Rey (1995), but one is particularly striking. Rey quotes Davidson as worrying about how to capture intentional causation of action as a species of practical inference brought about "in the right sort of way":

> A climber might want to rid himself of the weight and danger of holding another man on a rope, and he might know that by loosening his hold on the rope he could rid himself of the weight and danger. This belief and want might so unnerve him as to cause him to loosen his hold, and yet it might be the case that he never *chose* to loosen his hold, nor did he do it intentionally. (Davidson, 1980: 79)

Here we have causation and semantic coherence (understood broadly as I have indicated above), nevertheless the action causally comes about in the wrong sort of way. Here, intuitively, the set of properties of this belief/desire pair that would explain why the process makes semantic/practical sense is not the set of properties that would also be causally responsible for the ensuing behavior. Intuitively, we want the very same properties that would make the transitions come out coherent to be the ones that are also causally implicated in the state transitions (or the causation of the purposeful action). But what are those properties of states by virtue of which

we see them as "making sense"? Well, the answer is obvious. They are the logico-semantic properties of thoughts. This is not a contrived or tendentious demand that thinking is (at least sometimes) like that. What is good about Davidson's argument is that it shows that we do indeed have this intuition: namely, we do indeed distinguish actions, which would make perfect sense in a given context, that are brought about in the wrong way from the ones brought about in the right way. Apply this case to pure thought processes. The situation is exactly parallel. We do seem to care about how thoughts are caused in order for them to count as thinking in this strong sense – see below. Thinking involves tokening of thoughts that are causally brought about in the right sort of way. In a nutshell, there is a robust sense of 'thinking' according to which the very same properties of thoughts that explain the semantic coherence of a thought process, i.e. the logico-semantic properties, are to be causally implicated in the state transitions that constitute the process.

To be sure, there are other, less stringently characterized, thought processes that fall under the heading of 'thinking' not only in the ordinary folk parlance but also in cognitive psychology.[38] But whatever else may qualify as thinking, it is thinking in the above stringent but perfectly kosher sense that is used as an argument for LOTH.[39] LOTH is offered as a solution to this puzzle: how is thinking, conceived in this (strong) sense, possible? This is the problem of thinking.[40]

6.2. Sensitivity to Syntactic Structure: What Is It?

I want to be very clear about how postulating structure sensitive processes is supposed to solve this problem. What are the underlying principles that make it possible to solve the problem by postulating structure sensitive processes? Indeed, what exactly is structure sensitivity, so that it removes the mystery? I will show that it is precisely because the answer to these questions equally applies to connectionists' proposed solution – if it works – that they count essentially as a LOT solution, even though a potentially new and exciting one.

There are two sides to the story. Everybody knows that the logico-semantic properties of thought, I have mentioned above, that we intuitively take to be causally responsible in a thought process that we characterize as thinking proper can be captured non-semantically. At least this is the hope for the full range of different semantic domains in thought – the whole project of theoretical AI can indeed be seen to be the fulfillment of this hope. I won't elaborate this side of the story, since it is well known.[41]

Rather, I want to elaborate on the other side of the story; namely, *how exactly* are the processes supposed to be sensitive to the syntactic structure of representations? What is it about syntactic structure that buys us the mechanization of thinking? Again, I will generally ignore the problem of intentional content and talk as if representations were purely syntactically characterized entities.

We have to be very careful about distinguishing between two levels at which we may understand 'syntactic structure,' because it is in fact precisely in virtue of this two-level picture that formal systems are so important in the study of cognition.

At one level, we may conceive the syntactic/formal structure *qua* physically realized in representation tokens. As we will see more clearly in a moment, **D**-b requires syntactic structure at this level. In other words, **D**-b requires causal, and not "logical," structue sensitivity. This is the concrete sense of syntactic/formal structure. On the other hand, we may understand 'structure' more abstractly as, for instance, required by the abstract characterization of SL; i.e. at a level where no commitment to how it is to be concretely realized has yet been made.

This distinction is important, since it is precisely because syntactic structure abstractly understood can be exhibited or realized in concrete physical structure that we can bring abstract logical/semantic relations down to earth and make them subject to causal/physical processes. In other words, it is only to the extent to which we have a formally/syntactically regimented semantic domain that we can see how semantically coherent behavior can be obtained in a thoroughly physical/mechanical medium. The key to this feat is the two-level picture of syntax.

I believe that ignoring (or at least not being clear about) this two-level picture of syntax has been at the heart of a lot of confusion about the nature of syntactic properties and the role they are supposed to play in LOTH, and for that matter, in the Computational Theory of Mind (CTM).[42] Consider the following two claims often made in the computatiunalist/functionalist literature in the same breath without any warning as if they could both be true at the same level:

(S1) The syntactic properties (or, form) of a complex symbol are (metaphysically) determined by its computational (causal/functional) profile.

(S2) The computational (causal/functional) profile of a complex symbol is (metaphysically) determined by its syntactic properties or form.

Clearly these two claims can't both be true, in any interesting sense, at the same level. However, they both seem to be true and often claimed to be so without any clear indication about their status. How is this possible? The answer would remain a mystery without the two-level picture of syntax I described.

The sense in which (Sl) is true is the sense in which syntactic properties are multiply realizable, i.e., *qua* conceived at an abstract level, very much like the level at which I gave the abstract characterisation of SL. What ultimately guides this abstract characterization, of course, is in some loose sense whatever is captured in the formalization of a semantic "domain." In other words, since to regiment the semantic coherence of representational processes in terms of syntax is just to try to capture in non-semantic terms the role that representations play in the economy of thought processes, the syntactic properties postulated *ipso facto* mimic the semantic properties of representations. But this is to say that the syntactic properties are those properties that make the representations play a *certain role*, however they are realized. This role is, of course, what characterizes the semantic behavior of representations. But once the semantic domain is formalized, this role is captured

and type-individuated by the syntactic transformational rules like the ones specified in SL above. Hence the sense in which (S1) is true is given by the fact that syntactic properties are said to be *whatever properties* that make the physical symbol tokens (causally) behave in the system the way they do.

At this level, when we talk about conjunctions, disjunctions, conditionals, and their constituents such as their conjuncts, disjuncts, antecedents, consequents, etc. we are not interested in their particular realizations or shapes. We know how to process a representation if we know its logico-syntactic form or structure: if a representation is, say, a conjunction, we know we may derive any one of its conjuncts. This is how the derivation rules are specified in SL. This is surely structure sensitivity. But structure sensitivity understood at this abstract level is of no help for understanding the mechanization of thinking. For this we have to understand causal/physical structure sensitivity, not an abstract/logical one. In other words, we have to see the engineering principles of obtaining structure sensitivity at the level of physical realization. This is where structure sensitivity could be causal. And the sense of 'syntax' appropriate for this is given by (S2).

The sense in which (S2) is true is the sense in which syntactic properties are conceived *qua* realized or implemented in a physical/computational medium. In other words, when we talk about syntactic properties of symbols as determining their causal/functional role we are talking of them under a hypothesized physical instantiation mapping, i.e., *qua* mapped onto some physical state some of whose (quasi-)physical features satisfy the identity criterion for being a symbol token of a particular type. The case of specifying a concrete notation for SL is parallel (§4). Once the first phase of specifying atomic symbols is done, the concrete specification of modes of combination will determine what count as conjunctions, disjunctions, negations, etc. At this level, what makes a token expression, say, a conjunction, i.e. what makes it to have the syntactic property of being a conjunction, is literally its having certain quasi-physical properties. For example, anything that looks like or has the same shape as

$$(x \& y)$$

where *x* and *y* are any sentence will count as a conjunction. The concrete specification of modes of combination is what determines what counts as a syntactically structured token. And the way it does this is by producing tokens that have certain quasi-physical properties in virtue of which the tokens count as belonging to whatever syntactic type they do.

In physically realized computational devices, the symbols are the representational states of the device that satisfy certain identity criteria for being the symbols they are. A certain state token of the system has the property of being a conjunction in virtue of having certain physical properties that are predetermined in the engineering design of the system. There may not be one set of such properties for any one single syntactic form or property. There may be sets of them that are functionally equivalent to each other in so far as the functioning of the system is

concerned. But what is essential here is that it is in virtue of some such predetermined set of physical properties that a certain state of the system will count, say, as a conjunction. This point is absolutely essential to properly understand *causual* structure sensitivity. For it is because certain physical properties of state tokens are what make them, say, conjunctions that the mechanisms that process these tokens can be causally sensitive to their syntactic structure (i.e. to certain of their physical properties that make them count as conjunctions) and process them accordingly (i.e. in the way conjunctions are supposed to be processed). This is structure sensitivity in a very robust sense. The processor of the device is literally causally sensitive to the physical structure of the states. And since this physical structure is what encodes information about the syntactic properties of the state tokens, it can be exploited in causal processing.[43]

A processing mechanism needs at least two kinds of information about the symbol tokens it processes in structure sensitive ways. One is the syntactic type identity (or, logical form) of the token. The other is the information about its proper syntactic constituents, i.e. about the type identity of its constituents if there are any. This is necessary because most inference rules involve comparing the constituents of different symbols as to their identity or diversity. Take, for instance, the rule called Disjunctive Syllogism, labeled [DS] in SL. It says: given a disjunction and the negation of *one of its disjuncts*, derive the other disjunct. Given two sentences, it is not enough to know what their logical form is. Even if their logical form is "known" to the processor (say, it knows that one of them is disjunction), the processor needs to determine whether the other sentence is type-identical to any one of the disjuncts of the disjunction. Similarly for many other inference rules.

As a consequence, causal structure sensitivity requires that the complex symbol tokens must at a minimum encode the information about *two kinds* of syntactic property: the overall syntactic form and the type-identity of their particular syntactic constituents *qua* syntactic constituents. In concatenative notations, or physical symbol systems, the way to do this is to preserve the constituent tokens in the complex symbol itself. In this way, it is guaranteed that a suitably designed processing mechanism will be able to recover, from the complex symbol token, all the information it needs to process it according to its syntactic structure. In concatenative schemes, this syntactic structure is encoded in a very straightforward way by the physical structure of the complex token as dictated by the specific mode of combination.

The causal sensitivity to syntactic structure of symbol tokens, therefore, amounts to causal sensitivity to certain physical properties of symbol tokens that are made computationally relevant. And it is in virtue of these physical properties that a given token qualifies as belonging to a certain symbol type, therefore as having whatever syntactic properties the type is supposed to have. In a nutshell, then, it is the physical properties of symbol tokens made computationally relevant that drive the semantically coherent behavior of the system. Since these physical properties are what encode the two kinds of syntactic properties of the tokens (or more accu-

rately what make them *count as* the symbol tokens of a *particular* type they are), and since the syntactic properties mimic the semantic properties, we can solve the central problem of mechanization of thinking if we can devise processing mechanisms that would process the physical tokens on the basis of their Computationally Relevant Properties (CRPs). This is what makes structure sensitivity such an exciting discovery of this century after the works of Frege, Russell and Turing. Here is how Fodor puts the same idea:[44]

> You connect the causal properties of a symbol with its semantic properties *via its syntax*. The syntax of a symbol is one of its higher-order physical properties. To a metaphorical first approximation, we can think of the syntactic structure of a symbol as an abstract feature of its [geometric or acoustic] shape. Because, to all intents and purposes, syntax reduces to shape, and because the shape of a symbol is a potential determinant of its causal role, it is fairly easy to see how there could be environments in which the causal role of a symbol correlates with its syntax. It's easy, that's to say, to imagine symbol tokens interacting causally in virtue of their syntactic structures. The syntax of a symbol might determine the causes and effects of its tokenings in much the same way that the geometry of a key determines which locks it will open. (1987:18–9)

What is absolutely crucial is to notice that neither Fodor in the quotation nor I in the previous several paragraphs said absolutely anything about concatenation (except briefly as an *illustration*), i.e. about how exactly the syntactic properties need to be physically exhibited or realized in order to achieve causal structure sensitivity. But this is not surprising. For it should by now be obvious that all that is essential for obtaining causal structure sensitivity is to design systems with their appropriate CRPs, whether or not they are the properties arising out of a concatenative or explicit realization of syntactic structure.

But do the connectionists' attempts to obtain causal structure sensitivity conform to the above pattern I specified about what structure sensitivity essentially is, i.e. to the pattern of devising systems that would process representations on the basis of their CRPs? And, perhaps a more fundamental question is: can connectionist complex representations have CRPs that could encode all the syntactic information relevant for their direct processing in genuinely structure sensitive ways?

Let's go back for a moment to Chalmers' experiment and see how the apparent success of structure sensitive processing can be explained. We may remember that we have a bunch of compressed connectionist representations of passive English sentences obtained by using Pollack's RAAM network. The transforming network takes these as input and produces compressed representations of active sentences. The reverse process seems equally successful. The most important point, however, is that the transforming network somehow learns to process similarly structured representations in similar ways for which it was never trained. There really seems to be some sort of causal structure sensitivity here successfully obtained over non-concatenatively complex sentences. How is this explained?

Anyone with a bit of knowledge of the mathematics involved in the analysis of dynamical physical systems can guess how Chalmers' network works. I cannot go into a detailed analysis of the network here, but I can convey the idea which is in

fact simple.[45] What Pollack's network does is to locate all the vectors with identical "logical form" into a more or less homogeneous subspace in the multidimensional vector space defined for the network. In other words, the encoding of structurally similar representations proceeds by grouping them in one region of the high-dimensional vector space. That is the point of training the RAAM network. It is trained to locate, for instance, all the conjunctions in a particular subspace. The location of a certain vector in that subspace is, in a certain sense, the determinant of its form. And Chalmers' transformation network learns to treat vectors located in that subspace all in a similar fashion. The hidden units of the transforming network learns to detect the "shape" of the complex input representation as located in the multi-dimensional subspace reserved for, say, passive sentences, or conjunctions, etc., and treat them accordingly as it is taught to do. That is how it succeeds in generalizing over vectors for which it is not trained. Of course, this is no surprise, since what connectionist networks are particularly good at is exactly to map one vector onto another in any way you like. When this process is regimented through training according to whatever transformational regularities are to be obeyed, what you get is the holistic processing of complex representations according to their implicitly realized syntactic structure. In fact, my guess is that when a cluster analysis is performed on the hidden units of the transforming network, it can be seen that they divide their space and group the incoming patterns exactly according to the subdivisions of the encoding network, i.e. according to the distinct logical forms of representations.

The point I want to emphasize here is that the transformational ("computational") profile of a complex connectionist representation is determined by its location in the vector space reserved for those kinds of representations (e.g., conjunctions or active sentences, etc.). And this in turn is determined (within the context of an already trained network) by the specific numerical values of the vectorial representations at specific positions. There is a clear sense in which this is the "shape" of this kind of representations made computationally relevant, i.e., their particular shape determines their processing profile, and determines it causally if the network is physically realized. Let me dwell on this point a bit more explicitly.

Complex connectionist representations carry the information of their own syntactic structure, but differently than the way their concatenatively realized counterparts carry it.[46] In other words, the compressed connectionist symbols, as we have seen, have constituent syntactic structure according to the standards we have developed on the basis of SL above (§4). An implicitly structured vectorial symbol does have quasi-physical properties[47] on the basis of which it counts as the complex symbol type it is, because the way it is obtained, i.e. the way its constituents are combined together, guarantees that it belongs to a syntactically disjoint symbol system in such a way that its constituent and logical structure can be uniquely recovered from these quasi-physical properties. The situation, in fact, is quite parallel to the concatenatively realized syntactic structure: a complex token belonging to such a scheme has a certain set of physical properties that encode the syntactic

structure in such a way that makes complete recovery possible. In both cases, it seems, all the syntactic information necessary for the processor to process them in required ways is in the complex tokens themselves encoded by the physical properties of the tokens.

But how is the processing of this information possible in the connectionist networks? For holistic operations on connectionist compressed representations to be general and reliable, the compressed (implicit) syntactic structure of the representations should be available to the processing network which does the holistic transformations. This syntactic structure is encoded by the physical/numerical properties of the complex symbol token. What are these properties? As we have seen, these properties are the specific patterns of activation values of the units of distributed representations that determine their location in the vectorial space according to their syntactic/logical form. It is furthermore these properties that are supposed to be exploited in their holistic processing. The transforming network must be tuned to detect these properties and transform the symbols accordingly in a direct, holistic fashion. In this sense they are the computationally relevant properties of connectionist complex representations, since they drive the semantically syntactically significant behavior of the system.

In short, if there is any sense to be made of connectionist structure sensitivity, it must be along these lines, i.e., by picking out some CRPs of the compressed representational vectors (i.e., those physical/numerical properties of the vectors that encode the syntactic information) that are to be fed into the transforming network. Whatever specific values of such CRPs are, it should be clear that all that is needed is some such features of the vectors that will – if the network is physically realized – causally effect the processing of the network in a systematic and desired way. These properties constitute, in some well defined sense, as I have indicated above, the "shape" of the connectionist symbols that would causally determine their computational profile (just as Fodor himself says – see the quotation above). In other words, if the transforming network is processing the incoming connectionist complex representations in a completely reliable and general way, this must be because it somehow "knows" how to decode the syntactic information encoded by the quasi-physical/numerical properties of those very same representations even though they have only implicit syntactic structure. Otherwise, there is no sense to be made by what connectionists might mean by 'structure sensitivity.'

Now, in the light of this, consider the following "argument." If there are no CRPs involved in the actual process of holistic connectionist transformations, then the reliability of a network with which it systematically generalizes for structurally similar new inputs is a miracle. If there is such a property, however, then holistic transformations on compressed representations are simply a new and, I submit, very exciting way of obtaining causal structure sensitivity, hence they conform to the basic pattern of what is essential for LOT paradigm.

But, of course, successful holistic processing, as I tried to briefly and informally describe above, is not a miracle, not at least in such restricted domains and models.

In fact, all the heavy mathematical wizardry of connectionists is in the process of finding such properties that are increasingly more powerful. The analyses are at the level of what is sometimes called "subsymbolic" processing (this is, in fact, also true in designing concatenative machines), but the explicit aim in such analyses is to secure powerful and adequate symbolic processing capable of explaining exhibited *cognitive* regularities like systematicity at the *cognitive or representational level*, and explaining them essentially by satisfying **D**.

At the moment, however, we don't have a full-blown model that would do, with complete reliability and generality, structure sensitive processing holistically, i.e. without first decomposing the complex connectionist symbols into their constituents, in a completely formalized semantic domain like first-order logic. And whether we can ever have such a model is yet to be seen. There are formidable difficulties that need to be surmounted. But some of them can be seen at an intuitive level.

Since the particular syntactic constituents are not present in the connectionist representations, the information about their type identity (together with their logical form if they happen to be still complex) must be encoded in a more roundabout way by the physical/numerical properties of the representation token. However, under the pressure of (arbitrarily) long and diverse recursive symbol formation, it is very difficult to see how this information can be reliably encoded implicitly in such a way that you never lose the information that must be made available to the processing mechanism. In other words, is it possible to make the information about the type identity of all the particular constituents without literally preserving them in the complex representation itself? We have seen that there must at least be two types of information that the processor needs: one is about the type identity of the constituents of the complex representation, and the other is about its logical/syntactic form. Both are necessary because, as we have seen in the example of the derivation rules of SL, sensitivity to syntactic structure requires them. What is potentially the trouble maker for the connectionist complex representations is the first type of information, i.e. information about the identity of particular constituents. In physically instantiated representations, concatenation is a perfect way to guarantee that such information will never get lost. Sure enough, under the operating assumptions we have made before, in complex connectionist symbols you *do* have the information that is necessary to recover all the syntactic constituents and their forms, and this information is physically encoded in the complex representation itself (in fact, again, it is these properties that make it the very complex representation it is), but the question is whether the physical properties that encode *this* information can be the very CRPs themselves, i.e. the very properties that would causally drive the syntactic transformations *directly*, *holistically*.

I really don't know the answer to this question. The issue seems to be empirical, or at least, open to further investigation, empirical or otherwise. However, as I have said before, I don't need to know the answer to make my point, which is a conceptual one. Suppose that the difficulties can be surmounted, and holistic processing can

be scaled up, as many connectionists hope and expect, to do a serious explanatory job. What are we to say then? Are we to say that such connectionist models with holistic structure sensitive processing capabilities provide a radically new and different alternative to the LOT paradigm? I don't think we would say this. Rather, we would say, they are radically different ways of being LOT models simply in virtue of the fact that they satisfy the requirements of being in the LOT paradigm, namely, postulating representational systems and mechanisms that satisfy **D**.

We would say this because, as we have seen, the general principle involved in the satisfaction of **D**-b by a physical symbolic system is equally involved in the connectionists' proposal to satisfy it. Hence, the connectionist models in question are still within the confines of LOT paradigm, insofar as only satisfaction of **D**, and not any particular way of satisfying it, is essential for LOT. To repeat, what is essential in obtaining causal structure sensitivity is this: *certain physical/numerical properties that encode the syntactic constituent structure are directly made computationally relevant, i.e. are directly made the very ones that causally drive the syntactico-semantic behavior of the symbols,* hence, of the system. This is the general principle behind **D**-b, and as such, it is this principle that constitutes the solution to the problem of thinking.[48] How you realize this general principle is irrelevant precisely because it is this general principle itself, and nothing stronger as to how it is to be satisfied, is required by the traditional arguments for LOTH since it is just *this* that solves the problems – see below (§7). What is relevant is whether you genuinely realize it in such a way that the realizing system comes out as technically/empirically adequate for the explanation of law-like cognitive regularities to a psychologically respectable degree. But this latter issue seems to be an empirical one yet to be settled on the basis of further research, at least in the case of the kinds of the connectionist proposals considered here.

Certainly, these CRPs, the "shape" of the connectionist symbols, are radically different at some level of analysis from the "shape" of concatenatively realized symbols of, say, a PROLOG machine implemented in a conventional von Neumann machine. But from the perspective of a properly understood LOT, they all count as symbols in LOT, and the processes are properly called structure sensitive symbolic processes, because what counts is the reliable transformation of representations themselves: as long as representations are reliably handled in the desired way, any physical medium with its appropriate CRPs would in principle do from the classicist perspective.[49]

7. Conclusion

I do not mean to downplay the importance of non-concatenative connectionist models by saying that they are ultimately LOT models. On the contrary, I want to view them as very important and in many ways quite exciting contribution to the LOT paradigm if they can ultimately be made to scale up to do serious job. True enough, so far LOT models have always been identified with computational archi-

tectures that use concatenative representational schemes. But, if connectionists are right about the possibility of satisfying **D** non-concatenatively, then we should treat this finding as a significant contribution to the proper understanding of what the LOT architecture essentially involves: concatenation is not necessary to satisfy **D**.[50] In other words, if connectionists are right, then what we have is not a radically different paradigm threatening to overthrow LOT paradigm, but rather a radically different way of being a LOT model.[51] Why is this important in a way that goes beyond a verbal point?

Perhaps the best way to see that the very notion of LOT (hence, generally, the Computational/Representational Theory of Mind) cannot conceptually be (and ought not to have been) tied to concatenation is to consider the arguments historically offered for LOTH. We have seen them in the previous section. The postulation of a representational system that essentially satisfies **D** is justified as the best available explanation of the cited cognitive phenomena. But can these phenomena justify any further claim about how exactly **D** must be satisfied?

It was the need for an adequate explanation of a certain set of empirical phenomena, namely the law-like cognitive regularities like systematicity and productivity, and the need to solve what I have called the problem of thinking, that motivated the postulation of LOT in the first place. But when we see that the explanation essentially draws only on satisfying **D** (as I spelled it out) and not on any *particular* way (like concatenation) of satisfying it, insisting that the notion of LOT should essentially involve concatenative realization of **D** becomes unmotivated, because the very reasons that have historically prompted to postulate a LOT do not in and of themselves justify any further and stronger claim about how to physically realize it.

This is why connectionists could claim to be able to explain systematicity for instance: they claim to have satisfied **D** in their non-concatenative models. In other words, when it comes to the explanation of the cognitive regularities, what is doing the work is solely the satisfaction of **D** (as spelled out), and not any particular way of satisfying it. This is the reason why tying the notion of LOT essentially to the concatenative satisfaction of **D** would be unjustified and ought not to have been attempted. Let me therefore repeat: no arguments that have prompted to postulate a LOT in the first place could underwrite any further claim about how to satisfy/instantiate **D**, and if so, no further and stronger claim should be made about the essential nature of LOT.

It is surely true that LOTH conceived in this way is empirically weaker than LOTH understood as essentially tied to concatenation, or for that matter, as tied to non-concatenation. But LOTH has still plenty of empirical content especially when considered in its historical context, i.e. *vis-à-vis* its theoretical rivals like mentalistic associationism and eliminativist behaviorism, or even *vis-à-vis* any intentional realist theory that would treat mental states atomistically, i.e., any theory that is not committed to there being any syntactically complex representational brain states but that nevertheless aims to explain the same range of empirical

cognitive/behavioral phenomena. The Representational/Computational Theory of Mind has been at the foundational core of the so-called Cognitive Revolution in psychology. It is therefore absolutely essential to be clear about what it is and is not committed to, especially in the light of remarks we have been hearing increasingly more often these days about a Kuhnian paradigm shift brought about by connectionist research.

I conclude that to the extent to which they can satisfy **D** the models that are being developed by connectionists who took F&P's challenge seriously are still LOT models, however new and potentially exciting ones they might be at that, when the notion of LOT is rightly understood. Hence, their rejection of the first horn of the dilemma presented by F&P fails. However, F&M are mistaken too in their insistence on the alleged necessary connection between concatenation and LOT. Defending LOTH does not require and ought not to be tied to such a strong and unnecessary feature like concatenative realization of **D**. Reminding us of this, if nothing else, is the connectionist contribution as far as the proper understanding of the very idea of LOT is concerned.

8. Appendix: A Curious Objection

F&M, towards the end of their paper (1990) take up one issue apparently brought out by a reviewer that is directly relevant to our discussion so far. The reviewer asks:

> ...couldn't Smolensky easily build in mechanisms to accomplish the matrix algebra operations that would make the necessary vector explicit (or better yet, from his point of view, ...mechanisms that are sensitive to the imaginary components without literally making them explicit in some string of units)? (F&M, 1990: 201–2)

To which F&M respond in the following way:

> But this misses the point of the problem that systematicity poses for connectionists, which is not to show that systematic cognitive capacities are possible given the assumptions of a connectionist architecture, but to explain how systematicity could be necessary – how it could be a law that cognitive capacities are systematic – given those assumptions.
>
> No doubt, it is possible for Smolensky to wire a network so that it supports a vector that represents *a*R*b* if and only if it supports a vector that represents *b*R*a*; and perhaps it is possible for him to do that without making imaginary units explicit... The trouble is that, although the architecture permits this, it equally permits Smolensky to wire a network so that it supports a vector that represents *a*R*b* if and only if it supports a vector that represents *z*S*q*... The architecture would appear to be absolutely indifferent as among these options. (1990: 202)

The first thing to notice about this argument, as noted by Chalmers (1991), is that it proves too much. F&M grant that there exist theoretically non-problematic connectionist implementations of (concatenative) classical architectures. Now, any such connectionist implementation has to be wired up in some specific way in order to be an implementation. But given any such implementation, we may

always say with respect to it: it could have been wired up in a different way such that it could no longer support the classical architecture, and therefore, it could no longer explain how systematicity can be nomological. Hence, we could conclude, connectionist wiring up is absolutely indifferent as among architectures that guarantee nomological systematicity and the ones that do not.

This shows that we need to be very careful about which counterfactuals (nomological necessities) need to be explained in a principled way. F&M's question is: How could systematicity be necessary? This question is ambiguous. It may be demanding an architectural (synchronic) explanation, or an evolutionary (diachronic) explanation. There are plenty of signs that F&M intend the question in the former sense. What kind of mechanism (cognitive architecture) could make systematicity necessary? Their answer is: only those mechanisms that enforce concatenative compositionality. But we have seen that those connectionist models that enforce non-concatenative compositionality would also guarantee systematicity in the required sense. Connectionists offer a mechanism that, if wired up in the proper way, guarantees that if the organism can represent *a*R*b*, it can also represent *b*R*a*. That is what non-concatenative modes of combination of atomic symbols (like Gödel numbering system, Tensor Product Representations, the RAAM Architecture, and others) promise to offer.

Similarly for inferential systematicity: given the existence of a proper transformation network, it will by nomological necessity transform similarly (non-concatenatively) structured representations in formally similar ways. If this is right, then the question "how have they come to exist in cognitive organisms?" (or, "how has the brain come to be wired up to nomologically exhibit these cognitive regularities?") is a different one. It is, I take it, the business of evolutionary theory, or perhaps, developmental psychology, to answer this kind of diachronic question. In the second paragraph, F&M seem to sort of slip from the synchronic to the diachronic sense of the question. It is of course possible to wire up the connectionist networks quite differently. But given that there seem to exist a class of connectionist models having the potential to guarantee systematicity, saying that they could always be wired up differently does not do any good to the F&M's argument, because the same point applies to concatenative models: their set-up could always be changed so that they can represent *a*R*b* if and only if they can represent *z*S*q*, or for that matter if and only if they can represent "The Last of The Mohicans." This kind of tinkering with the architecture does not count and is outside the rules of the game.[52]

Notes

1. Smolensky, for instance, is explicit in his rejection of this horn: "...distributed connectionist architectures, without implementing the Classical architecture, can nonetheless provide structured mental representations and mental processes sensitive to that structure" (1990a: 215).
2. I will present what these cognitive regularities are and how they constitute arguments for LOTH below in §6.1 along with some additional arguments.

3. Although I will examine it in more detail below, for the moment, concatenative/explicit tokening or realization is roughly one in which the syntactic constituents of a complex representation token are literally present in the complex token itself: they are literally part of the syntactically complex expression. By contrast, a non-concatenatively or implicitly realized syntactic structure is where constituents are not explicit in this sense. We will see below some of the connectionist ways in which this can be done.
4. As I will talk occasionally, an implicit or non-concatenative satisfaction of **D** is one where the syntactic structure is implicit or non-concatenatively realized in the complex representation and the structure sensitivity is obtained without making the syntactic constituents explicit. Similarly, *mutatis mutandis*, for explicit/concatenative satisfaction of **D**.
5. Namely, *if* connectionist representations have syntactic (and semantic) structure, and mechanisms processing such representations are sensitive to syntactic structure, *then* connectionist models are implementation models of LOT architecture. For the refutation claim, see, for instance, Chalmers (1990, 1991).
6. I don't know anyone who has a logical reading in mind for the impossibility claim, although Rey (1995) comes pretty close.
7. I will touch upon some of the difficulties involved in structure sensitive progessing of implicitly structured representations below.
8. Robert Cummins has criticized Pylyshyn's (1984) notion of functional architecture, which is more or less the same notion as the one under consideration here, and proposed a more specific notion of cognitive architecture: "Pylyshyn often makes it sound as if the primitive operations of a programming language define a functional architecture, but this cannot be right. The functional [cognitive] architecture of the mind is supposed to be that aspect of the mind's structure that remains fixed across data structures (i.e., in what is represented). This is the [hardwired] program itself, including its control structure, not the primitive operations of a language we might write in" (1989:165–6). I think that Cummins is right about this. Although the difference between these two conceptions will not be very important and I will generally have Cummins' more specific notion in mind for what follows, it is worth emphasizing that Cummins' remarks bring out one potentially confusing issue clearly. Namely, when we talk about the architecture, we are talking about the mechanisms and their organization rather than the representations or data structures over which these mechanisms operate. On the other hand, when F&P talk about **D** as the defining characteristics of classical architecture, their emphasis is on the nature and format of representations, rather than the mechanisms that operate on them. This may be potentially confusing, but need not be. See below.
9. Consider, for instance, the basic architecture of Marvin Minsk's (1967) simplest universal Turing machine with only four symbols and seven intrinsic states. A first-order theorem prover can in principle be implemented in it. But in such a case, the primitive operations (there are only twenty-eight of them!) cannot be defined over "representational" states (i.e., over the interpreted well-formed formulas as such), since the four kinds of symbols cannot individually be used representationally; rather, their combinations would have to serve as "representational" states of the virtual theorem prover. See below.
10. There is no necessity that the formation rules of a formal language be recursive or combinatorial. But since the most interesting formal systems, and more importantly, the ones we are interested in (**D**-a requires combinatorial rules) have such languages, I will have in mind for what follows only formal systems that incorporate recursive/combinatorial formation rules.
11. In their article, it is not clear what notion of 'implementation' F&P had in mind when they put their dilemma against connectionists. But the textual evidence suggests that they use 'implementation' in the technical sense I specified above, since they seem to assume that an implementation model must use the network nodes and/or connections non-representationally in implementing the structured representations of the higher-level classical architecture. But merely satisfying **D** is not implementation in this sense. Unfortunately, missing this important point has generated all sorts of confusion in the literature. Smolensky (1988, 1990a), however, is one of the few people who is aware of the problem and the distinction. He claims that in the technical sense of 'implementation' his tensor product representations can't be implementation of **D**-a. He is surely

right about this. But the conclusion he draws, namely, that his brand of connectionism is a radical alternative to LOT, surely doesn't follow.

12. A general and technically elaborate description of the basic architecture of tensor product systems can be found in Smolensky (1990b). See Smolensky (1990a) for an informal and easily accessible discussion of the same issues. For a truly impressive application of the tensor product technique to higher cognitive processes, see Smolensky (1995).
13. This claim is problematic. However, it is generally assumed to be true in the literature. I will continue to pretend that it is true, since, in a certain sense, my aim is to work out its philosophical consequences if it were true.
14. See, for instance, Hinton (1990), Elman (1989), St. John and McClelland (l990) among others.
15. McLaughlin (1993a) repeats the accusation against Chalmers (1990) who uses in his experiment (see below) Pollack's RAAM architecture to produce vectorial representations.
16. The discussion that follows in this section owes a great deal to Michael Devitt and Georges Rey (in conversation), Smolensky (1990a, 1995) and especially to van Gelder (1990). I should also cite in this connection the pioneering works of Goodman (1976) and Haugeland (1982) that helped me to sort out many difficult issues here. See also Devitt (1990) and Goel (1991) for similar helpful discussions. Although there are various points over which I disagree with some of these authors, I take my discussion to be complementary to, and not in competition with, theirs. And, of course, the conclusion I draw from my discussion is just the opposite of what Smolensky and van Gelder draw.
17. This is to accommodate in an informal way the extra complication created by the difference between operations that accept one and two arguments.
18. For an interesting attempt to characterize first-order predicate logic without a commitment to a particular notation, see Thomason (1969). In my abstract characterization of SL I do not claim to have captured every aspect of SL that a logician might want to be very curious or scrupulous about. My aim is just to convey the basic idea.
19. Moreover, just for the sake of completeness: the identity criterion for a type must be such that it be not only *theoretically* but also *actually* (practically/technologically) possible for something to satisfy it, and that not everything satisfy it.
20. Insisting on paying attention to what is involved in the identification of computational states and their causal transitions that would preserve the type-identity of the states is an essential part of answering the often heard claim that anything can be described to compute any function! For a nice elaboration of an answer along this line, see Goel (1991). See Searle (1984, 1992) and Putnam (1988) for versions of the claim.
21. It would be an interesting exercise to show what sorts of properties could figure in providing identity criteria for types. Since tokens are necessarily spatio-temporal particulars, there is a loose sense in which the identifying properties would be "physical," whatever exactly that means. Trying to get clear about this loose sense involves complications however. Shapes, for instance, are usually cited as physical properties, or "higher-order" physical properties. But shapes can be multiply realized without, it seems, being functionally defined or definable. Shapes of letters, for instance, can be realized in a variety of physical media: think of letter 'A' inscribed in sand, wax, etc. In this sense, shapes still seem to be abstract entities. At any rate, I will sometimes use 'quasi-physical' to indicate the looseness of the sense of 'physical' in 'physical properties of symbol tokens.' I should, however, note that in physically realized dynamical symbol systems (like computers and possibly human brains) the sense of 'physical' must *ipso facto* be quite robust and straightforward since it is these properties that are partly responsible for causally driving the state transitions of the system in time.
22. Cf. Goodman (1976) and van Gelder (1990).
23. Cf. van Gelder (1990) and Fodor and McLaughlin (1990). I should, however, note that F&M don't use the term 'concatenation,' their preference is to use 'Classical' instead!
24. Note that many actual physical realizations of abstract formal systems like von Neumann computers are also concatenative in just this sense: when such a conventional computer stores, for instance, a token of a well-formed complex expression of its machine language in many of its registers equipped with a pointer system, the registers literally contain tokens of its constituents, albeit in a spatially distributed fashion.

25. See especially Chalmers (1990) and van Gelder (1990). Butler (1991) appeals approvingly to Chalmers (1990). Smolensky's insistence (1990a, 1995) that symbolic processing emerges out of node-level subsymbolic processing in Tensor Product models is in fact meant to be a claim that these models can process connectionist complex representations in a structure sensitive fashion. Smolensky's discussion, however, contain some curious and strange elements that make it hard to follow his reasoning. He claims that processing of tensor product representations, even though structure sensitive, can only be explained at a node-level analysis, and he says, it is partly this feature that makes his model different from classical models. But, this can't make the Tensor Product Model non-classical, since the same is true for instance for a first-order LISP theorem prover run in a very simple von Neumann machine: there is a certain sense in which the "really" causally efficacious elements are to be found in the hard-wired organization of the actual von Neumann machine that implements the LISP machine. If so, we can equally say that the real explanation (where real causation occurs, so to speak) is at the organizational level of particular von Neumann architecture where the data processes are no longer "classical-cognitive." I think that saying this would be wrong. See F&M (1990) for a criticism of Smolensky on this point.
26. The reason I am saying 'could' is that in Chalmers' experiment we might have only a fragment of a possibly integrated connectionist system. If this kind of approach turns out to be successful – it is by no means obvious that it will, see below – it is reasonable to expect that more serious models will have more integrated and complex architectures consisting of many subnetworks. Chalmers' experiment shows some of the basic principles about how some connectionists propose to handle structure sensitive processes. This is *all* I want to show with Chalmers' model.
27. In his experiment, Chalmers made no attempt to capture tense and noun-verb agreement in active-passive transformations. McLaughlin (1993a) attacks Chalmers by rightly claiming that it was precisely these difficulties that led Chomsky to postulate a "deep structure" from which active and passive forms can be obtained. So he accuses Chalmers of false advertisement: Chalmers' model is not a successful connectionist model that can adequately explain English active-passive transformations. I think that this criticism is right but not quite relevant here. In fact, it is unfortunate that Chalmers had chosen to model this particular phenomenon in order to *illustrate* how connectionist models can handle structure sensitive operations holistically, thus explain inferential systematicity. All McLaughlin shows is that the very structure Chalmers had chosen in order to illustrate how it could be causally used in structure sensitive processing happened to be the wrong kind of structure. But nothing really should hang on this. He could have illustrated holistic processing on the transformation rules of SL for instance. The point is whether structure sensitivity *can* be achieved by holistic transformations. I would certainly agree with McLaughlin if his claim were that Chalmers' experiment doesn't show that such connectionist models can be scaled up to do full-blown structure sensitive processing to a psychologically respectable degree. But, again, my point is that it *illustrates* nicely some of the ways in which connectionists *might* handle structure sensitivity.
28. Compare the following remark by F&P: "If you hold the kind of theory that acknowledges structured representations, it must perforce acknowledge representations with similar or identical structure... So, if your theory also acknowledges mental processes that are structure sensitive, then it will predict that similarly structured representations will generally play similar roles in thought" (1988: 48).
29. There are problems about the proper description of what formative systematicity is supposed to be. I think that it is no accident that all attempts to describe systematicity at some point appeal to using examples. But it is not clear what exactly, at the end, examples succeed at conveying in the way of what is alleged to be a thoroughly pervasive law-like regularity about the cognitive economy of certain organisms. If systematicity is to be used as an argument for LOTH, it must be describable as an empirical phenomenon without any implicit or explicit appeal to a **D**-like structure. Otherwise the argument for LOTH and against connectionism would be circular. The problem is how to do this without using any examples. Now, of course, the use of examples may be kosher at some stage, but then, if it can't be eliminated without risking circularity at the end, it is not clear what facts might constitute empirical counterexamples to systematicity.
30. Rey (1991) describes eight cognitive phenomena whose explanation, he claims, requires LOT, and hence, they are arguments for LOTH. All the eight phenomena are cast out as certain features

or regularities of having propositional attitudes. Again, all of them but the last one are formative regularities in the sense I am using the term.

31. I should, however, say that I am not confident even about this claim. For, obviously, just putting different ideas together will not be enough to explain the semantic unity of the content of *judgments*. You have to put the ideas together in the right way, i.e. in such a way that would explain, for instance, why judging that Mary loves John is different from judging that John loves Mary even though they are both put together out of the same ideas, i.e. out of the ideas of John, Mary, and loving. But putting ideas in the right way in a representational medium/format seems to require syntax in its most natural sense. You have to be able to tell a story about why the two judgments are different by saying that the constituent ideas are, as F&P put it, "in different construction" with each other. But this seems to amount to appealing to syntactic structure. British empiricists' story seems to be appropriate not in explaining *judgments*, but rather, at best, in explaining the acquisition or construction of complex *concepts* on the basis of simple ones. For an insightful and revealing discussion of parallel issues within the context of connectionist modeling, see F&P (1988:15–32) and Rey (1991).
32. Fodor in the Appendix of (1987) gives another argument he calls "methodological" for LOTH. It is designed to infer the existence of structurally complex internal brain states from the structural complexity of actions they cause. This argument also appeals to a species of what I have called inferential cognitive regularity: namely, the structured nature of actions, i.e. structurally complex behavior *caused* by intentional states. For an explicit incorporation of action *qua* intentional process to thinking as such, see Davies (1991).
33. For most lucid and explicit statements of this kind of argument for LOTH, see Fodor (1985) and (1987: 12–14 and 1987: Appendix); Haugeland (1985); Rey (1995).
34. Indeed Fodor's excitement and enthusiasm in this respect cannot be overstated: "The real achievement is that we are (maybe) on the verge of solving a great mystery about the mind: *How could its mental processes be semantically coherent?* Or, if you like yours with drums and trumpets: How is rationality mechanically possible? Notice that this sort of problem can't even be stated, let alone be solved, unless we suppose... that there are mental states with both semantic contents and causal roles" (1987: 20). Fodor's point is that *syntactically structured symbols* physically realized in the brain are the only things that can fill those roles.
35. Contrary to the supposition of some like Searle (1984) and Putnam (1988), the Computational Theory of Mind (i.e. the theory – more or less – that spells out how **D**-b is supposed to work) does not offer a solution to the problem of how it is possible to have intentional states – Fodor's first question. For Fodor's attempts to solve this problem, see his (1987) and (1991).
36. Davies (1991), Rey (1995). Cf. Lycan (1993) who gives a "deductive" argument for LOTH on the basis of considerations about the productivity of thought.
37. See Fodor (1980, 1985, 1987) (see especially Fodor's discussion of why Conan Doyle was a better psychologist than James Joyce or even Willian James in 1985:10–24 – the discussion is somewhat reiterated in 1987:12–4); F&P (1988); Davies (1991); Rey (1995).
38. Let me emphasize one point here. There is a very strong sense in which the problem of thinking can be construed as the problem of logical inference in that reasoning is in conformity to the cannons of formal logic like validity or derivability. This is not the sense in which I characterize thinking here. As an anonymous referee rightly pointed out, this is a highly contrived characterization, not shared by many psychologists. What I have in mind is a weaker sense: *whatever semantic coherence* belongs to certain forms of thinking they all arise out of processes that are formally specifiable. This doesn't entail that so specified thinking conforms to cannons of logic. As I pointed out, validity would be the ideal of semantic coherence, but it is not necessarily the relevant property to be captured formally. This is why theoretical AI has been investing so heavily into non-classical logics. This is important to keep in mind because the very puzzle LOT is supposed to solve arises from the intuition that local mental causation must be reducible to local non-semantic properties of brain states, which relates to Fodor's third puzzle I quoted above.
39. Here I am obviously assuming that there is a real psychological phenomenon corresponding to our folk concept of thinking conceived in this strong sense. Not that all our "thinking" is like this, but that we at least sometimes engage in this kind of thinking. There are people who deny

this. I don't want to address this issue here. See Rey (1995) for more discussion. Suffice it to say that the connectionists who take F&P's challenge seriously and develop models to meet it seem to agree that thinking is indeed something like this, since they attempt to meet the challenge in basically the same sort of way, namely, by postulating structured sensitive processes, albeit non-concatenatively obtained. One reason why those connectionists who reject to meet the challenge are not moved by such traditional arguments for LOTH is that many of them reject as somehow spurious the cognitive phenomena whose adequate explanation, classicists claim, requires LOT. Hence they reject the characterization of thinking given here. As I set aside the discussion of the other horn of F&P's dilemma at the start, I cannot go into the evaluation of the prospects of such a move here. But see McLaughlin (1993b) for more discussion.

40. The argument from thinking when understood in the above way is connected to the argument from inferential systematicity because the two phenomena are connected in a deep way. Thinking requires that the logico-semantic properties of a *particular* thought process (say, inferring that John is happy from knowing that if John is at the beach then John is happy and coming to realize that John is indeed at the beach) be somehow causally implicated in the process. The systematicity of inferential thought processes then is based on the observation that if the agent is capable of making *that* particular inference, then she is capable of making many other somehow *similarly organized* inferences. But the idea of similar organization in this context obviously demands some sort of a classification of thoughts independently of their *particular* content. But what can the basis of such a classification be? The only basis seems to be the logico-syntactic properties of thoughts. Although I am not comfortable about talking of syntactic properties of thoughts common-sensically understood, it seems that they are forced upon us by the very attempt to understand their semantic properties: how, for instance, could you explain the semantic content of the thought that if John is at the beach then he is happy without somehow appealing to its being a *conditional*? This is the point of contact between the two phenomena. When, especially, the demands of naturalism are added to this picture, inferring to a LOT (= a representational system satisfying **D**) realized in the brain becomes indeed almost irresistible.
41. The rough outline goes something like this. Modern logic has taught us that the behavior of semantic properties can be studied non-semantically, i.e. proof-theoretically, where this means roughly, syntactically. And the rise of modern computers has shown that whenever the behavior of any semantic domain can be formalized, i.e., syntactically captured, we can build physical devices, usually called computers, which would exhibit the same behavior, i.e. devices whose state transitions would mimic the behavior of the semantic domain. Hence, that the brain is such a computational device that operates on syntactically structured representations is roughly the LOTH.
42. See Devitt (1990) and (1996) for a helpful and somewhat similar discussion of syntax. Stich's Syntactic Theory of Mind (1983) (STM) was based on the notion of syntax based on (Sl), and as such, it was a purely functionalist view. Missing this point has caused in the literature a quick and ultimately mistaken assimilation of the STM to the CTM and was at the source of many unfortunate confusions that prevented people from seeing what was really wrong with the STM, for an examination of which see my (1995a).
43. As far as we are clear about what we are talking about, using terminology like "a physical property encoding syntactic information" is harmless. Once the identity criteria are given for type-individuation of symbol tokens, anything that satisfies a given criterion will be a symbol of that particular type, not just that it will carry the information about its type-identity.
44. I should add, however, that in general Fodor is among the least careful in the use of 'formal/syntactic' despite the fact that he makes heavy use of this notion in crucial ways in different parts of his entire theoretical corpus sometimes with disasterous and very confusing results. For a criticism of Fodor on the notion of syntax, see my (1995b).
45. What follows is an informal analysis of Chalmers' network. I didn't perform an actual numerical analysis of it myself. Networks like Pollack's and Smolensky's are known to be dynamical state space systems whose state transitions can be explained in terms of differential equations. I am simply assuming here that the same sort of analysis is true of Chalmers' transforming network and am using it as an illustration. If this assumption may turn out to be false, as an anonymous referee pointed out that it might, then there is no point in trying to squeeze into such

connectionist models a reading whereby they turn out to satisfy **D**-b. But then given the lack of any alternative understanding how such models successfully perform their assigned tasks and generalize to new data, the connectionists in question have no basis whatsoever to deny the first horn of F&P's dilemma, that is, as I will indicate below, there is no legitimate sense in which there are connectionist models that may satisfy **D**. Also, since I am using Chalmers' network as an example, if it turns out that its success is accidental, the framework used in the analysis of connectionist networks I assume here is not thereby disqualified.

46. Again, let me remind the reader that this way of putting the point is just a shorthand for saying that the way connectionist representations *are* syntactically structured is different from the way explicitly structured representations have syntactic constituent structure. In general, we can perhaps say: if a representation has syntactic structure then it carries the information of its syntactic structure. And here I want to focus on the information carried.
47. In simulations, these properties are usually numeric, but the idea is that in physical realizations, these numeric values will correspond to genuine physical quantities like voltage level, firing frequency, pulse rate and intensity, etc.
48. It is at least an essential part of the solution in the sense that whatever the ultimate specific story turns out to be about thinking the general principle will be true of it.
49. See Cummins (1989) and Cummins and Schwartz (1991) for a somewhat parallel discussion and conclusion.
50. Since Turing, it so happened that all the interesting physical computers we have actually built or designed happened to use concatenative symbolic schemes. Part of the connectionists' contribution then might be seen to lie in the fact – if it is a fact – that this was a historical accident and there was nothing metaphysically necessary about it. This could hardly be a trivial result. What I am suggesting is that the historical association of LOT architecture with the kind of concatenative machines traditionally used in AI may have conditioned people to think of LOT paradigm always in these terms, namely essentially requiring a concatenatively realized symbolic language. What I am urging therefore is that this is not essential about LOT.
51. In a way, we may even classify the historically traditional Concatenative LOT models as Classical-LOT (C-LOT) models and the Non-Concatenative connectionist ones as NonClassical-LOT (NC-LOT) models. But both kinds would still be LOT models. Shortly after this paper got out of my hands back to the editor of this journal, an article by T. Horgan and J. Tienson came to my attention ("Structured Representations in Connectionist Systems?" Steven Davis, ed., *Connectionism: Theory and Practice*, Oxford, UK: Oxford University Press, 1992) in which the authors make a similar claim after a discussion somewhat similar to the one I presented here.
52. I would like to thank many people for their support, encouragement and help while I was writing this paper. I am especially grateful for their insightful comments and criticisms to Ken Aizawa, David Chalmers, Jon Cohen, Michael Devitt, Güven Güzeldere, Jesse Prinz, Georges Rey, Philip Robbins, Brian Smith, and Ken Taylor. I would also like to thank John Perry and the CSLI crowd in Stanford for their warm hospitality and never ending help during my stay there as a visiting scholar while struggling with the issues I discuss here. Also, many thanks to the audiences of the talks I gave in Stanford University, University of Maryland and the University of Chicago at which I presented different sections of this paper.

References

Aydede, Murat (1995a), "Computation and Functionalism: Can Psychology Be Done 'Syntactically'? to appear in *Boston Studies in the History and Philosophy of Science*, Dordrecht: Kluwer.

Aydede, Murat (1995b), "On the Type/Token Relation of Mental Representations," *Facta Philosophica* Vol. 2, No. 1, pp. 23–49, March 2000.

Butler, Keith (1991), "Towards a Connectionist Cognitive Architecture," *Mind and Language* 6, No. 3, pp. 252–72.

Chalmers, David (1990), "Syntactic Transformations on Distributed Representations," *Connection Science* 2, pp. 53–62.

Chalmers, David (1991), "Why Fodor and Pylyshyn Were Wrong: The Simplest Refutation," in *Proceedings of the l2th Annual Conference of the Cognitive Science Society*, pp. 340–7.

Cummins, Robert (1989), *Meaning and Mental Representation*, Cambridge, Massachusetts: The MIT Press.

Cummins, Robert and Georg Schwarz (1991), "Connectionism, Computation, and Cognition," in Terence Horgan and John Tienson, eds., *Connectionism and the Philosophy of Mind*, Studies in Cognitive Systems (Volume 9), Dordrecht: Kluwer Academic Publishers, 1991.

Davidson, Donald (1980), "Freedom to Act," *Essays on Actions and Events*, Oxford: Oxford University Press.

Davies, Martin (1991), "Concepts, Connectionism, and the Language of Thought," in W. Ramsey, S.P. Stich and D.E. Rumelhart, eds., *Philosophy and Connectionist Theory*, New Jersey: Lawrence Erlbaum.

Devitt, Michael (1990), "A Narrow Representational Theory of the Mind," In W.G. Lycan, ed., *Mind and Cognition*, Oxford: Basil Blackwell.

Devitt, Michael (1996), *Coming to Our Senses*: *A Naturalistic Program for Semantic Localism*, Cambridge, UK: Cambridge University Press.

Elman, Jeffrey L. (1989), "Structured Representations and Connectionist Models," in *Proceedings of the Eleventh Annual Meeting of the Cognitive Science Society*, Ann Arbor, Michigan, pp. 17 – 23.

Fodor, Jerry A. (1975), *The Language of Thought*, Cambridge, MA: Harvard University Press.

Fodor, Jerry A. (1980), "Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology," *RePresentations: Philosophical Essays on the Foundations of Cognitive Science*, Cambridge, Massachusetts: The MIT Press, 1981. (Originally appeared in *Behaviorial and Brain Sciences* 3, 1, 1980.)

Fodor, Jerry A. (1985), "Fodor's Guide to Mental Representation," *A Theory of Content and Other Essays*, Cambridge, Massachusetts: The MIT Press, 1990.

Fodor, Jerry A. (1987), *Psychosemantics: The Problem of Meaning in the Philosophy of Mind*, Cambridge, Massachusetts: The MIT Press.

Fodor, Jerry A. (1991), "Replies" (Ch.15), in B. Loewer and G. Rey eds., *Meaning in Mind: Fodor and His Critics*, Oxford: Basil Blackwell, 1991.

Fodor, Jerry A. and Zenon W. Pylyshyn (1988), "Connectionism and Cognitive Architecture: A Critical Analysis," in S. Pinker and J. Mehler, eds., *Connections and Symbols*, Cambridge, Massachusetts: The MIT Press (A Cognition Special Issue).

Fodor, Jerry A. and B. McLaughlin (1990), "Connectionism and the Problem of Systematicity: Why Smolensky's Solution Doesn't Work," *Cognition* 35, pp. 183–204.

Goel, Vinod (1991), "Notationality and the Information Processing Mind," *Minds and Machines* 1, pp. 129–165.

Haugeland, John (1982), "Analog and Analog," in J.I. Biro and R.W. Shahan, eds., *Mind, Brain and Function*, Oklahoma: University of Oklahoma Press.

Haugeland, John (1985), *Artificial Intelligence: The Very Idea*, Cambridge, Massachusetts: The MIT Press.

Goodman, Nelson (1976), *The Languages of Art*, Second Edition, Indianapolis, Hackett.

Hinton, Geoffrey (1990), "Mapping Part-Whole Hierarchies into Connectionist Network," *Artificial Intelligence* 46, Nos. 1–2 (Special Issue on Connectionist Symbol Processing).

St. John, M.F. and J.L. McClelland (1990), "Learning and Applying Contextual Constraints in Sentence Comprehension," *Artificial Intelligence* 46, Nos. 1–2 (Special Issue on Connectionist Symbol Processing).

Loewer, Barry and Georges Rey (eds.), (1990), *Meaning in Mind: Fodor and His Critics*, Oxford: Basil Blackwell.

Lycan, William (1993), "A Deductive Argument for the Representational Theory of Thinking," *Mind and Language*, Vol. 8, No. 3, pp. 404–22.

McLaughlin, B.P. (1993a), "The Connectionism/Classicism Battle to Win Souls," *Philosophical Studies* 71, pp. 163–90.

McLaughlin, B.P. (1993b), "Systematicity, Conceptual Truth, and Evolution", in C. Hookway and D. Peterson, eds., *Philosophy and Cognitive Science*, Royal Institute of Philosophy, Supplement No. 34.

Minsky, M. (1967), *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice Hall.

Pollack, J.B. (1990), "Recursive Distributed Representations", *Artificial Intelligence*, Vol. 46, Nos. 1–2 (Special Issue on Connectionist Symbol Processing), November 1990.

Putnam, Hilary (1988), *Representation and Reality*, Cambridge, Massachusetts: The MIT Press.

Pylyshyn, Zenon W. (1984), *Computation and Cognition: Toward a Foundation for Cognitive Science*, Cambridge, Massachusetts: The MIT Press.

Rey, Georges (1991), "An Explanatory Budget for Connectionism and Eliminativism," in T. Horgan and J. Tienson, eds., *Connectionism and the Philosophy of Mind*, Dordrecht, Kluwer Academic Publishers, 1991.

Rey, Georges (1995), "A Not "Merely Empirical" Argument for a Language of Thought" in J. Tomberlin, ed., *Philosophical Perspectives* 9, pp. 201–222.

Searle, John R. (1984), *Minds, Brains and Science*, Cambridge, MA: Harvard University Press.

Searle, John R. (1992), *The Rediscovery of the Mind*, Cambridge, Massachusetts: The MIT Press.

Smolensky, Paul (1988), "On the Proper Treatment of Connectionism," *Behavioral and Brain Sciences* 11, pp. 1–23.

Smolensky, Paul (1990a), "Connectionism, Constituency, and the Language of Thought," in B. Loewer and G. Rey, eds., *Meaning in Mind*: *Fodor and His Critics*, Oxford: Basil Blackwell, 1991.

Smolensky, Paul (1990b), "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems," *Artificial Intelligence*, Vo1. 46, Nos. 1–2 (Special Issue on Connectionist Symbol Processing), November 1990.

Smolensky, Paul (1995), "Reply: Constituent Structure and Explanation in an Integrated Connectionist/Symbolic Cognitive Architecture," in Cynthia and Graham Macdonald, eds., *Connectionism: Debates on Psychological Explanation*, Oxford: Basil Blackwell.

Stich, Stephen P. (1983), *From Folk Psychology to Cognitive Science: The Case Against Belief*, Cambridge, Massachusetts: The MIT Press.

Thomason, R.H. (1969), *Symbolic Logic*, New York: Macmillan.

van Gelder, Timothy (1990), "Compositionality: A Connectionist Variation on a Classical Theme," *Cognitive Science*, Vo1. 14, pp. 355–384.

van Gelder, Timothy (1991), "Classical Questions, Radical Answers: Connectionism and the Structure of Mental Representations," in Terence Horgan and John Tienson, eds., *Connectionism and the Philosophy of Mind*, Studies in Cognitive Systems (Volume 9), Dordrecht: Kluwer Academic Publishers, 1991.